



UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
CENTRO DE CIÊNCIAS AGRÁRIAS E ENGENHARIAS
CURSO DE GRADUAÇÃO EM ENGENHARIA QUÍMICA

NOME DO AUTOR 1

NOME DO AUTOR 2

...

TÍTULO AQUI

Alegre - ES

ANO

NOME DO AUTOR 1

NOME DO AUTOR 2

...

TÍTULO AQUI

Trabalho de Conclusão de Curso apresentado ao Curso de Engenharia Química do Centro de Ciências Agrárias e Engenharias da Universidade Federal do Espírito Santo, como requisito parcial para avaliação da disciplina de Trabalho de Conclusão de Curso I (ou Trabalho de Conclusão de Curso II).

Orientador(a): Nome do Orientador

Co-orientador(a): Nome do Coorientador

Alegre - ES

ANO

TÍTULO AQUI

NOME DO AUTOR 1

NOME DO AUTOR 2

...

Trabalho de Conclusão de Curso apresentado ao Curso de Engenharia Química do Centro de Ciências Agrárias e Engenharias da Universidade Federal do Espírito Santo, como requisito parcial para avaliação da disciplina de Trabalho de Conclusão de Curso I (ou II).

Aprovado em DIA de MÊS de ANO.

Prof.(a) Nome do Orientador, D.Sc.
Universidade Federal do Espírito Santo

Prof. Nome do Coorientador, D.Sc.
Universidade Federal do Espírito Santo

Prof.(a) Membro Interno, D.Sc.
Universidade Federal do Espírito Santo

Engenheiro Químico Membro Externo
Universidade X

AGRADECIMENTOS

É um item opcional.

Pode ser usado para agradecer aos colaboradores à execução do trabalho.

Agradecer à FAPES, CAPES, . . . pelo apoio financeiro por meio da bolsa concedida.

Se desejar, preencha apenas no TCC II

RESUMO

ÚLTIMO SOBRENOME DO AUTOR, Nome do autor por extenso. **TÍTULO AQUI**. ANO. Trabalho de Conclusão de Curso (Bacharelado em Engenharia Química) - Universidade Federal do Espírito Santo, Alegre, ES. Orientador: Nome do Orientador. Co-orientador: Nome do Coorientador.

O resumo é um elemento obrigatório e é elaborado de acordo com a NBR 6028. Todo o resumo deve ser feito em apenas um parágrafo, sem recuo de primeira linha, fonte Arial 12, entrelinha 1,5. Deve-se usar o verbo em voz ativa e na terceira pessoa do singular. O resumo deve conter de 150 até 500 palavras. No resumo, são ressaltados o objetivo da pesquisa, o método utilizado, as discussões e os resultados com destaque apenas para os pontos principais. O resumo deve ser significativo, composto de uma sequência de frases concisas, afirmativas, e não de uma enumeração de tópicos. Não deve conter citações. Na próxima linha, devem ser colocadas as palavras-chave (palavras ou expressões significativas retiradas do texto), precedidas do texto "Palavras-chave:", em negrito, sem marcação de parágrafo e separadas uma das outras por vírgula (,) e finalizadas com um ponto (.). As palavras-chaves são representativas do conteúdo do documento, devendo ser escolhidas preferencialmente em vocabulário técnico. Evitar repetição de palavras do título. São grafadas com as iniciais em letra minúscula, à exceção de substantivos próprios e nomes científicos.

Palavras-chave: Palavra 1, Palavra 2, Palavra 3.

ABSTRACT

ÚLTIMO SOBRENOME DO AUTOR, Nome do autor por extenso. **TÍTULO EM INGLÊS.**
ANO. Final Course Assignment. (Bachelor degree in Chemical Engineering) – Federal
University of Espírito Santo, Alegre, ES. Adviser: Nome do Orientador. Co-adviser:
Nome do Coorientador.

Consiste efetivamente na tradução do item RESUMO para a língua inglesa, seguido
de "Keywords". Todas as normas expostas anteriormente são válidas para este item.
Sugere-se utilizar alguma ferramenta de IA, como o chatGPT ou o Gemini, para auxiliar
na tradução ou adaptação do resumo para a língua inglesa.

Keywords: Word 1, Word 2, Word 3.

LISTA DE FIGURAS

Figura 1 – Resultado da pesquisa: <i>Livro Métodos numéricos em problemas de engenharia química</i>	21
Figura 2 – Pop-up da ferramenta de citação do Google.	21
Figura 3 – Coloque na descrição da imagem um texto que explique ao leitor o que ele está vendo na imagem, por exemplo: Gráfico do volume de reclamações recebidas (em azul) e resolvidas (em vermelho) ao longo dos meses.	24
Figura 4 – Capa do livro “Métodos Numéricos em Problemas de Engenharia Química” (Pinto; Lage, 2001).	26
Figura 5 – Exemplo de como exportar uma tabela do Google para o formato \LaTeX usando a extensão <i>Spread-Latex</i>	28
Figura 6 – Logomarca da UFES.	42

LISTA DE QUADROS

Quadro 1 – Exemplo de quadro usando um figura.	29
Quadro 2 – Exemplo de quadro usando uma tabela.	30

LISTA DE TABELAS

Tabela 1 – Exemplo de tabela	27
Tabela 2 – Tabela criada no Planilhas do Google usando a extensão Spread-LaTeX	29

LISTA DE ABREVIATURAS E SIGLAS

ABNT	Associação Brasileira de Normas Técnicas
TCC	Trabalho de Conclusão de Curso
pH	Potencial Hidrogeniônico de íons de hidrogênio (H ⁺)
PID	Controlador Proporcional-Integral-Derivativo
MPC	Controle Preditivo por Modelo
IMC	Controle por Modelo Interno
PET	Poli(tereftalato de etileno)
PVC	Poli(cloreto de vinila)

LISTA DE SÍMBOLOS

CO_2	Dióxido de Carbono
C_{3+}	Hidrocarbonetos com três ou mais carbonos
K_p	Ganho estacionário do processo
Γ	Letra grega Gama
Λ	Lambda
ζ	Letra grega minúscula zeta
\in	Pertence
k_t	Constante de terminação
Nu	Número de Nusselt
Pe	Número de Peclet
Pr	Número de Prandtl
Re	Número de Reynolds
Sh	Número de Sherwood

SUMÁRIO

1	INTRODUÇÃO	14
2	OBJETIVOS	15
2.1	Objetivo geral	15
2.2	Objetivos específicos	15
3	JUSTIFICATIVA	16
4	REVISÃO BIBLIOGRÁFICA	17
4.1	Inserindo citações	17
4.1.1	Citações Indiretas	18
4.1.2	Citações Diretas	18
4.1.3	Citação de Citação	20
4.1.4	Apresentação de autores no texto	20
4.1.5	Código da Referência	21
4.2	Inserindo figuras	22
4.3	Inserindo tabelas e quadros	25
4.3.1	Tabelas	26
4.3.2	Quadros	29
4.4	Inserindo equações	30
4.4.1	Equações matemáticas	30
4.4.2	Equações químicas	32
4.4.2.1	Usando o pacote <code>chemfig</code>	33
4.4.2.2	Usando o pacote <code>mhchem</code>	35
4.5	Notas de Rodapé	36
5	MATERIAIS E MÉTODOS	37
6	RESULTADOS	38
7	CRONOGRAMA	39
	REFERÊNCIAS	40
	APÊNDICE A – LOGOMARCA DE UFES	42

APÊNDICE B – NULLAM ELEMENTUM URNA VEL IMPERDIET SODALES ELIT IPSUM PHARETRA LIGULA AC PRETIUM ANTE JUSTO A NULLA CURABITUR TRISTIQUE ARCU EU METUS	43
ANEXO A – REFERÊNCIAS RÁPIDAS EM \LaTeX	44
ANEXO B – CHEMFIG: UM PACOTE PARA DESENHAR MOLÉCULAS \LaTeX	47
ANEXO C – MORBI ULTRICES RUTRUM LOREM.	131
ANEXO D – CRAS NON URNA SED FEUGIAT CUM SOCIIS NATORQUE PENATIBUS ET MAGNIS DIS PARTURIENT MONTES NASCETUR RIDICULUS MUS	132
ANEXO E – FUSCE FACILISIS LACINIA DUI	133

1 INTRODUÇÃO

No contexto acadêmico, a produção de trabalhos científicos, em especial os Trabalhos de Conclusão de Curso (TCC), demanda uma organização rigorosa, tanto no conteúdo quanto na apresentação visual. Por isso, um modelo padronizado, isto é, um *template*, bem estruturado desse tipo de documento torna-se uma ferramenta essencial. Dessa forma, os estudantes podem contar com uma base sólida para a elaboração de seus trabalhos.

Este *template* de TCC em \LaTeX vem preencher uma lacuna importante nesse processo, fornecendo uma estrutura pronta e formatada de acordo com as normas acadêmicas vigentes. Ao utilizar este modelo, vocês ganham tempo e tranquilidade, pois podem focar mais na produção do conteúdo sem se preocupar excessivamente com questões de formatação.

Além disso, o \LaTeX apresenta várias vantagens em relação a outros editores de texto, especialmente quando se trata de documentos acadêmicos. Sua sintaxe simples e intuitiva permite uma formatação consistente e profissional, garantindo uma apresentação visual de alta qualidade. A capacidade de lidar facilmente com equações matemáticas, referências cruzadas e bibliografias também o torna ideal para trabalhos científicos. Além disso, a integração com plataformas online de compartilhamento e colaboração, como o Overleaf, adiciona um elemento extra de conveniência e eficiência ao processo. Diante desses benefícios, este *template* se apresenta como uma escolha inteligente e eficaz para estudantes que buscam produzir trabalhos acadêmicos de qualidade, com praticidade e profissionalismo.

Neste capítulo, serão apresentados exemplos práticos de como inserir os elementos típicos de um trabalho de conclusão de curso (ou mesmo, de uma dissertação ou tese). Serão abordados procedimentos para inclusão de figuras, tabelas, equações, citações e outros elementos relevantes. O \LaTeX é uma ferramenta altamente versátil, oferecendo diversas maneiras de adicionar esses elementos ao documento. São demonstradas abordagens mais simples e acessíveis para iniciantes. Com essas instruções, espera-se proporcionar uma base sólida o suficiente para que você possa compor seu trabalho de forma eficiente e satisfatória.

2 OBJETIVOS

Este capítulo delinea os objetivos gerais e específicos deste documento, fornecendo uma visão sucinta do propósito e das metas a serem alcançadas.

2.1 Objetivo geral

O objetivo geral é fornecer um modelo padronizado em LaTeX para Trabalhos de Conclusão de Curso (TCC) para o curso de Engenharia Química do Campus de Alegre da UFES, de modo a facilitar a organização e formatação rigorosa exigida pela produção acadêmica, permitindo aos estudantes concentrarem-se mais na produção de conteúdo.

2.2 Objetivos específicos

Os objetivos específicos são:

1. Apresentar um *template* pré-formatado para o TCC, seguindo as normas acadêmicas vigentes, visando proporcionar uma estrutura pronta e formatada para os estudantes.
2. Destacar as vantagens de um documento escrito em \LaTeX em relação a outros editores de texto, enfatizando sua capacidade de fornecer uma formatação consistente e profissional, que inclui facilidade na inserção de equações matemáticas, referências cruzadas e bibliografias.
3. Apresentar exemplos práticos de como inserir elementos típicos de um TCC, como figuras, tabelas, equações e citações, fornecendo abordagens simples e acessíveis, especialmente direcionadas a iniciantes.
4. Proporcionar instruções que permitam aos estudantes compor seus trabalhos de forma eficiente e satisfatória, fornecendo uma base sólida para a elaboração do conteúdo acadêmico.

3 JUSTIFICATIVA

A produção de trabalhos científicos, especialmente o TCC, é uma etapa crucial na formação acadêmica, permitindo aos estudantes consolidar e aplicar os conhecimentos adquiridos ao longo de seu percurso educacional. No entanto, a elaboração desses trabalhos requer não apenas um sólido embasamento teórico e metodológico, mas também uma apresentação visual e estrutural que atenda às exigências das normas acadêmicas vigentes.

A utilização de um modelo padronizado, como o template em \LaTeX proposto, se justifica pela necessidade de fornecer aos estudantes uma ferramenta que simplifique e otimize o processo de formatação de seus trabalhos. Ao contar com uma estrutura pré-formatada de acordo com as normas acadêmicas, os estudantes podem direcionar seus esforços para a produção de conteúdo de qualidade, sem se preocupar excessivamente com questões de formatação que podem consumir tempo e recursos.

Além disso, a escolha do \LaTeX como plataforma para a elaboração desses trabalhos é justificada pela sua capacidade de proporcionar uma apresentação visual consistente e profissional. A sintaxe simples e intuitiva facilita a formatação de elementos complexos, como equações matemáticas e referências cruzadas, garantindo uma apresentação de alta qualidade que atenda aos padrões exigidos pela comunidade acadêmica. Assim, ao adotar este template e utilizar o \LaTeX como ferramenta de produção, os estudantes podem elevar o nível de seus trabalhos acadêmicos, contribuindo para a qualidade e relevância da pesquisa científica.

4 REVISÃO BIBLIOGRÁFICA

O capítulo de revisão bibliográfica desempenha um papel fundamental na fundamentação teórica do trabalho acadêmico, fornecendo uma análise crítica e abrangente das principais fontes e estudos relevantes para o tema em questão. Este capítulo não apenas contextualiza o problema de pesquisa dentro do corpo existente de conhecimento, mas também identifica lacunas, contradições e áreas de debate que justificam a realização do estudo proposto.

Ao revisar a literatura pertinente, o objetivo é fornecer ao leitor uma compreensão sólida do estado atual do campo de estudo, destacando teorias, conceitos-chave e descobertas importantes. A revisão bibliográfica não se limita apenas a resumir trabalhos anteriores, mas também deve sintetizar e analisar criticamente as abordagens, métodos e resultados encontrados em estudos relevantes, proporcionando assim uma base sólida para o desenvolvimento do argumento do trabalho.

Neste capítulo, serão exploradas as técnicas e práticas fundamentais para a inserção eficaz de elementos acadêmicos essenciais no documento. Em particular, será abordada a inserção de citações, destacando métodos adequados para citar fontes e referenciar estudos relevantes dentro do texto, conforme as normas de citação adotadas. Além disso, serão apresentadas estratégias para a inserção de tabelas e quadros, essenciais para a apresentação organizada de dados e informações, fornecendo diretrizes claras sobre como formatar e referenciar esses elementos adequadamente.

A seguir, será discutida a inserção de figuras, ilustrações e gráficos, elucidando técnicas para integrar visualmente esses elementos ao texto de maneira coerente e informativa, com ênfase na legibilidade e compreensão dos dados apresentados. Por fim, serão exploradas as formas de inserção e formatação de equações matemáticas, essenciais para trabalhos científicos que envolvem análises quantitativas, com instruções detalhadas sobre como incorporar equações ao texto.

4.1 Inserindo citações

Uma parte essencial de qualquer trabalho acadêmico são as citações e referências. A citação é a menção, no texto, de uma informação colhida de outra fonte. Pode ser *direta*, *indireta* e *citação de citação*. Nas citações, as chamadas pelo sobrenome do autor, pela instituição responsável ou título incluído na sentença devem ser letras maiúsculas e minúsculas (ver norma atualizada: ABNT 10520: 2023).

4.1.1 Citações Indiretas

É a transcrição livre do texto do autor consultado. As citações indiretas ou parafraseadas dispensam o uso de aspas duplas e do número de páginas.

Exemplos:

- A otimização de plantas industriais pode se beneficiar com a integração de *design* automático nos principais simuladores de processos (Minasidis; Jäschke; Skogestad, 2013).
- Sabe-se que McDonagh *et al.* (2024) destacaram que uma classe importante de tecnologia comercial de captura de carbono envolve o uso de solventes.

Para fazer essas duas referências, podem-se usar os seguintes comandos:

- Primeiro, ao escrever um texto com suas próprias palavras baseado na obra de um autor, insere-se o comando `\cite{codigo_da_referencia}` no ponto onde deseja citar a obra consultada. Essa citação virá entre paratênses.
- A outra maneira é iniciar uma frase fazendo referência ao autor com o comando `\citeonline{codigo_da_referencia}`. Essa citação deverá ser integrada à frase, devendo se atentar ao tempo verbal e número de pessoas.

Com isso, essas formas podem ser usadas de modo criativo:

- É necessário consolidar um arcabouço transdisciplinar de fundamentos, teorias e metodologias para solucionar as questões emergentes (van Gigch; Pipino, 1986).
- Os parâmetros utilizados neste trabalho já foram bastante explorados na literatura (Schwaab *et al.*, 2008).
- De acordo com Minasidis, Jäschke e Skogestad (2013), o conceito é . . .
- Como responder à questão levantada por Wilson e Madsen (2010)?

4.1.2 Citações Diretas

Citações diretas são reproduções textuais de trechos escritos por outros autores e devem ser usadas quando a alteração do texto original poderia distorcer seu significado ou quando se deseja mencionar diretamente as palavras de um autor. Por isso, esse tipo de citação é a transcrição literal de palavras ou trechos de autores.

Elas podem ser classificadas em *curtas*, com menos de 3 linhas, e devem ser integradas ao corpo do texto entre aspas duplas, e *longas*, com mais de 3 linhas, as quais devem

ser apresentadas com recuo abaixo do texto. Para referenciar a um trabalho neste tipo de citação, são usados novamente os comandos `\cite{codigo_da_referencia}` e `\citeonline{codigo_da_referencia}`.

As citações curtas devem ser escritas entre ‘‘FRASE’’`\cite{codigo_da_referencia}`. As aspas são formadas com crase dupla e encerradas com aspas simples duplas, seguido da referência ao trabalho que contém a frase.

No caso da citação direta longa, deve ser criada uma ambiente com a sintaxe:

```
\begin{citacao}
  As citações diretas, no texto, com mais de três linhas [...]
  deve-se observar apenas o recuo
  \cite[p. num_pagina]{codigo_da_referencia}.
\end{citacao}
```

Exemplos:

- Citação curta

Conforme destacado por [Dellamatrice e Monteiro \(2014, p. 3\)](#), “os ecossistemas aquáticos de rios possuem grande heterogeneidade física, que incluem de pequenos a grandes rios, variação no regime de vazão e geomorfologia do canal, entre outras características do habitat.”

- Citação longa

Quanto aos principais aspectos da poluição de rios brasileiros por pesticidas, é importante destacar que:

Alguns fatores que afetam o transporte para o meio aquático são as propriedades do agente químico e variáveis ambientais, como tipo de solo, declividade, presença de cobertura vegetal e clima; alguns desses fatores, como relevo planáltico e solo quartzoso, estão presentes em grande parte do país fazendo com que a poluição do meio aquático seja acentuada. ([Dellamatrice; Monteiro, 2014, p. 1](#))

Em ambos os casos, quando o trecho citado não for início de parágrafo, deverá ser antecedido de reticências entre colchetes [...]. Caso contrário, se o texto citado for interrompido antes do ponto final do parágrafo, deverá ser precedido de reticências entre colchetes [...].

Além das supressões mostradas acima, podem ser feitas interpolações, acréscimos, comentários usando [], e também ênfases ou destaques usando sublinhado ou negrito ou itálico). O ponto final deve ser usado para encerrar a frase e não a citação.

4.1.3 Citação de Citação

É citação direta ou indireta de um documento ao qual não se teve acesso ao original. Os elementos devem ser indicados na seguinte ordem: autoria ou a primeira palavra do título; data; página do documento original, se houver; a expressão apud; autoria ou a primeira palavra do título; data; página da fonte consultada, se houver.

Na lista de referências, elencar somente a fonte consultada. Para isso, a entrada do artigo que não foi lido deve receber o marcador @hidden no arquivo *.bib.

- De acordo com [Silva et al. \(2008\)](#) apud [Baptestini et al., 2015](#), p. 1204), na secagem em leito de espuma (*foam-mat drying*), alimentos líquidos ou semilíquidos são transformados em espumas estáveis usando agitação intensa e aditivos.
- Devido ao tipo de região de confiança, o método de verossimilhança é normalmente preferido ([Donaldson; Schnabel, 1987](#) apud [Schwaab et al., 2008](#), p. 1543).

Neste caso, as referências de [Silva et al. \(2008\)](#) e de [Donaldson e Schnabel \(1987\)](#) não podem estar presentes na lista de referências, porque não foram obras consultadas.

4.1.4 Apresentação de autores no texto

Deve-se obedecer aos seguintes critérios:

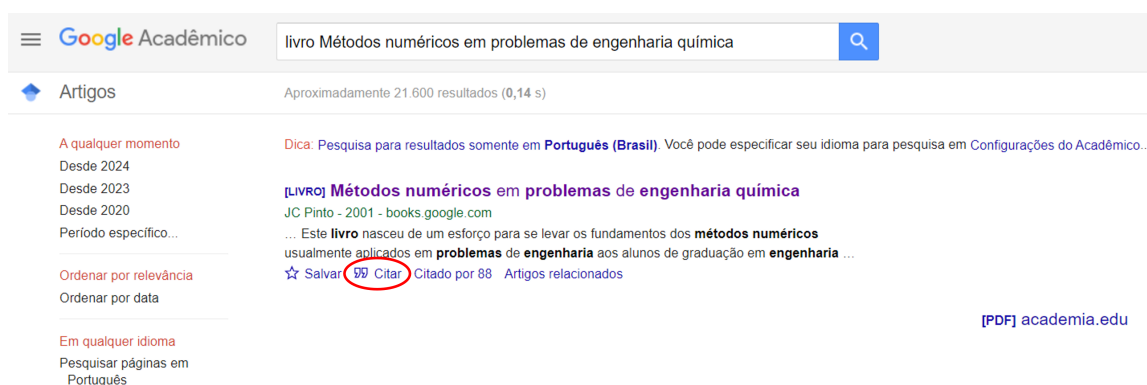
1. **Um autor** - indicação do sobrenome do autor, seguido da data:
[Guizzardi \(2005\)](#) demonstrou que ...
 O uso de modelo conceituais ([Guizzardi, 2005](#)) é cada vez mais explorado.
2. **Dois autores** - quando a autoria da obra for dois autores, todos serão citados:
 O trabalho de [Wilson e Madsen \(2010\)](#) é um ponto de partida para ...
 A codificação foi documentada há bastante tempo ([Wilson; Madsen, 2010](#)).
3. **Três autores** - quando a indicação da fonte consultada estiver inserida no texto, os nomes dos autores serão interligados por vírgula e os dois últimos interligados pela conjunção “e”. Se a indicação da fonte consultada estiver entre parênteses serão separados por ponto-e-vírgula:
 Em [Minasidis, Jäschke e Skogestad \(2013\)](#), foi visto que ...
 O sistema de controle são complexos ([Minasidis; Jäschke; Skogestad, 2013](#)).
4. **Mais de três autores** - indicação do primeiro autor, seguido da expressão *et al.* (em itálico) acrescidos da data e número da página. Na lista de referências, deve constar todos os autores:
[McDonagh et al. \(2024\)](#) apresentaram uma nova representação ...
 A capacidade de absorção de CO_2 é maior que 80% ([McDonagh et al., 2024](#)).

4.1.5 Código da Referência

Se você olhar na lista de arquivos, vai encontrar um último arquivo chamado *Referencias.bib*. Esse é um arquivo que precisa ser alimentado com suas referências no formato/código *BibTex*. Conseguir esse código não é complicado, basta usar, por exemplo, o Mendeley ou o Google Scholar (<https://scholar.google.com/>).

Suponha que se queira usar uma referência ao livro “Métodos Numéricos em Problemas de Engenharia Química” dos professores José Carlos Pinto e Paulo Lage. Então basta fazer a pesquisa no Google Scholar e clicar no item apresentado na Figura 1.

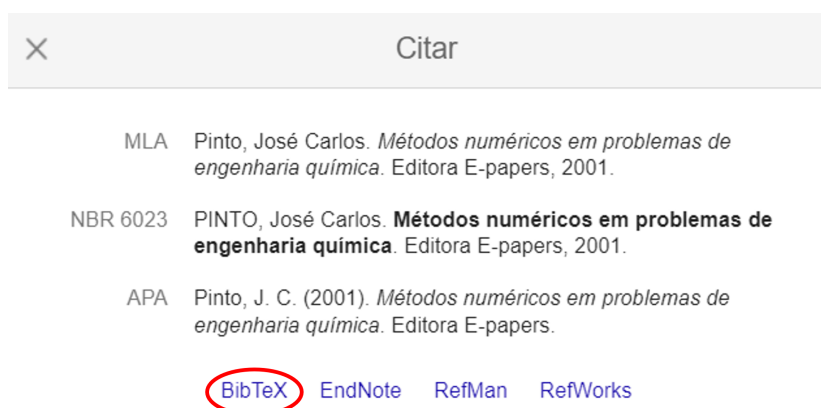
Figura 1 – Resultado da pesquisa: *Livro Métodos numéricos em problemas de engenharia química*.



Fonte: Autoria própria.

Ao clicar no *link* circulado em vermelho, você verá a Figura 2 que é uma ferramenta do Google para citações nas mais variadas ferramentas de citação.

Figura 2 – Pop-up da ferramenta de citação do Google.



Fonte: Autoria própria.

Em vermelho você vê destacado a opção de mostrar a referência no formato Bib-tex. Ao clicar neste link, você verá o código, que deve ser copiado para o arquivo textitReferencias.bib:

```
@book{pinto2001metodos,
  title={Métodos numéricos em problemas de
  engenharia quântica},
  author={José Carlos Pinto and Paulo Laranjeira
  da Cunha Lage},
  year={2001},
  publisher={Editora E-papers}
}
```

Você pode aprender no Google o que significa cada item do BibTex funciona. O código da referência é o que está logo após a chave na primeira linha. No caso do livro do pesquisado, o código da referência é **pinto2001metodos**. Esse código é uma sugestão do Google, você pode editar essa parte para um código que seja mais fácil de memorizar. Além disso, esse código tem que ser único, não pode ter duas referências com o mesmo código.

Quando for necessário fazer uma referência a esse livro, basta usar:

- `\cite{pinto2001metodos}`, ou
- `\citeonline{pinto2001metodos}`


Por exemplo:

- O método de volumes finitos é frequentemente confundido com o método das diferenças finitas devido ao fato de que, em muitos casos, as equações discretizadas resultantes de ambos os métodos são idênticas, conforme dito por [Pinto e Lage \(2001\)](#).
- Sabe-se que as “[...] técnicas de natureza local usualmente requerem um grande número de pontos de discretização, para que a solução final obtida aproxime bem a solução real procurada” ([Pinto; Lage, 2001](#), p. 100).

4.2 Inserindo figuras

As imagens devem estar centralizadas horizontalmente na página com tamanho da fonte de 10 pt. As figuras, bem como tabelas e quadros, podem ser inseridos no corpo do texto ou na forma de anexos/apêndices. Se forem inseridos no texto, devem aparecer o mais próximo possível de onde forem mencionados. Estes elementos devem estar centralizados na página, sua identificação na parte superior e a indicação da fonte consultada (elemento obrigatório, mesmo que seja produção do próprio autor), legenda, notas e outras informações necessárias à sua compreensão (se houver). Os que forem

citações diretas dever ter o número da página, e as que forem citações indiretas, deve ser da seguinte forma “Fonte: [Minasidis, Jäschke e Skogestad \(2013\)](#), adaptado pelo autor”. Ou modificado, ou traduzido.

No plataforma do *Overleaf*, é muito simples inserir uma figura. Use o pacote `graphicx`, que é padrão na maioria das distribuições LaTeX. Este pacote permite inserir imagens em diferentes formatos, sendo aconselhável trabalhar sempre em *PNG*, *JPG* ou *PDF*. Você precisa fazer o *upload* da imagem para o Overleaf clicando no ícone , que fica no canto superior esquerdo, e seguir as instruções.

Dicas de cuidado na organização dos arquivos:

- Use nomes que deem significados aos arquivos. Evite usar "figura1.jpg", "figura2.jpg" etc. Ao longo da sua escrita, muitas figuras serão adicionadas e retiradas e essa numeração vai criar um caos na hora que você precisar modificar alguma coisa. Por exemplo, o nome do arquivo do ícone de upload que inseri no texto é *icone-upload.png*;
- Organize os seus arquivos em pastas. Conforme seu trabalho cresce, o número de arquivos também e pode ser um caos achar uma imagem que precisa ser substituída. Eu já deixei uma estrutura pré-organizada, dentro da pasta Arquivos, há subpastas para anexos, apêndices e figuras. Se achar melhor, pode criar mais subpastas, por exemplo, para organizar as figuras por capítulos.

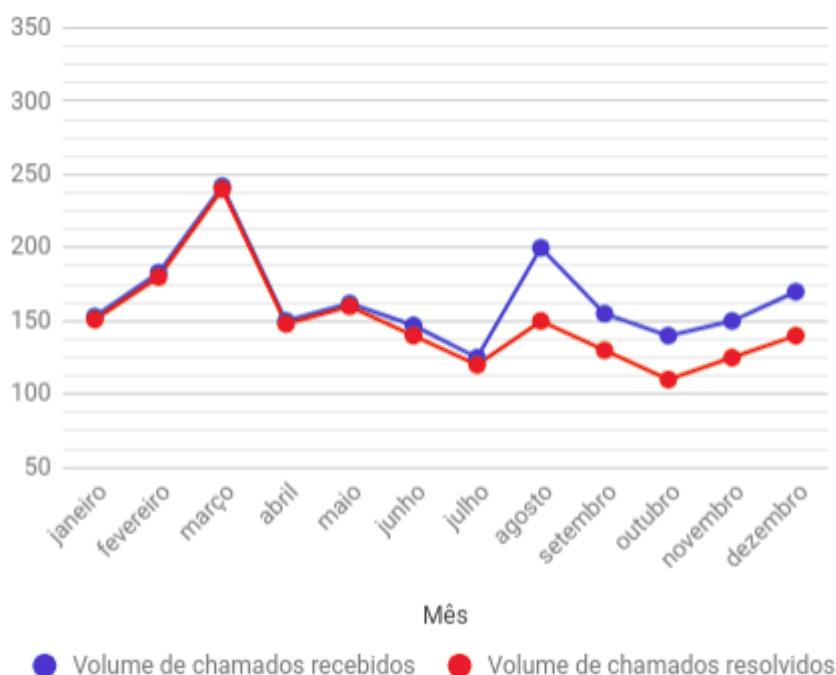
A Figura 3 foi adicionada utilizando o código:

```
\begin{figure}[p]
  \footnotesize % tamanho 10pt
  \caption{Coloque na descrição da imagem um texto que explique
    ao leitor o que ele está vendo na imagem, por exemplo:
    Gráfico do volume de reclamações recebidas (em azul) e
    resolvidas (em vermelho) ao longo dos meses.}
  \label{fig:volume}
  \centering
  \includegraphics[scale=0.5]{Figuras/volume_recebidos.png}
  \fonte{\citeonline{Alura}}
\end{figure}
```

Observe que adicionar um objeto no \LaTeX terá sempre uma estrutura que irá se repetir:

1. Começa com um `\begin{figure}` e termina com o `\end{figure}`. Isso é o que chamamos de iniciar (begin) e encerrar (end) um ambiente de um objeto. No caso, o objeto é uma figura (figure);
2. Um item de descrição da imagem `\caption{Descrição da imagem}`. Repare que a descrição não precisa definir o número da imagem, o \LaTeX faz isso automaticamente para você;
3. Um item de rótulo de imagem `\label{fig:volume}`. Esse rótulo serve para facilitar a referência do objeto. A qualquer momento do texto eu vou poder escrever `\ref{fig:volume}` e o \LaTeX vai saber que a que objeto você se refere e colocar automaticamente o número do objeto;
4. A tag `\centering` é para dizer ao \LaTeX que centralize a imagem na página. É importante lembrar que no \LaTeX você não deve se preocupar com o posicionamento dos objetos, ele faz isso automaticamente para você;
5. Vai aparecer o objeto propriamente dito, sendo que a imagem é adicionada com o código: `\includegraphics[scale=0.5]{Figuras/volume_recebidos.png}`.
Veja que há uma variável *scale* que permite modificar o tamanho da imagem, e que é introduzida a imagem através do caminho de onde você a salvou;

Figura 3 – Coloque na descrição da imagem um texto que explique ao leitor o que ele está vendo na imagem, por exemplo: Gráfico do volume de reclamações recebidas (em azul) e resolvidas (em vermelho) ao longo dos meses.



Fonte: Sampaio (2011)

6. Pela norma ABNT, figuras, tabelas e quadros precisam ter suas fontes citadas. Portanto, mesmo se a autoria for sua, deve aparecer o item `\fonte{Do autor}` ou `\fonte{Autoria própria}` para creditar a fonte. Seja consistente com a forma escolhida.

Vale destacar que junto à criação do ambiente da figura, pode-se definir as preferências de posicionamento da figura no documento, fazendo

```
\begin{figure}[opções]
  % Conteúdo da figura
\end{figure}
```

Substitua a palavra *opções* pelas preferências de posicionamento desejadas, sendo as mais comuns:

- `[h]`: Posiciona a figura "aqui"(onde ela é definida no código).
- `[t]`: Posiciona a figura no topo de uma página.
- `[b]`: Posiciona a figura na parte inferior de uma página.
- `[p]`: Coloca a figura em uma página separada, dedicada exclusivamente a figuras.
- `[!htbp]`: Força o LaTeX a posicionar a figura exatamente onde você a define no código, sem levar em consideração outras considerações de formatação.

Você pode combinar essas opções conforme necessário, como `[ht]` para preferir a posição "aqui" se possível, mas aceitar o topo - a primeira letra tem maior prioridade. Se nenhuma opção for especificada, o LaTeX tentará posicionar a figura da melhor forma possível com base em suas próprias regras de formatação.

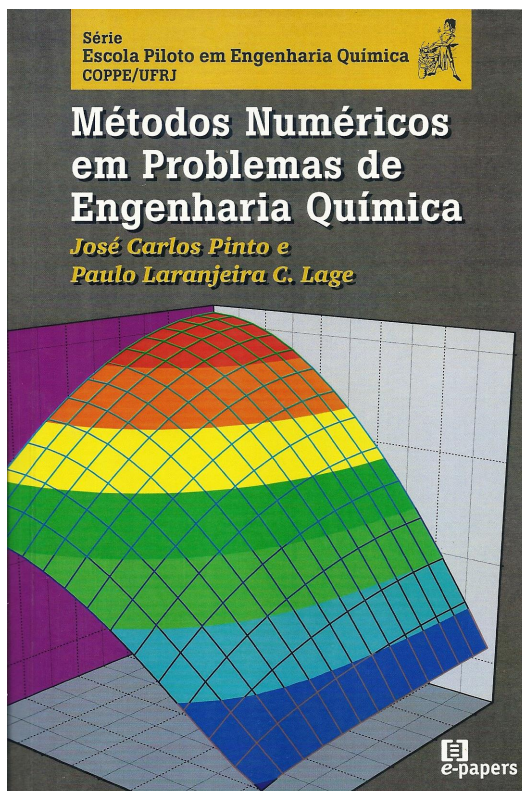
Como mais um exemplo, veja a Figura 4. A chamada desta figura no texto é feita com `\ref{fig:capalivro}`, com a numeração ajustada automaticamente pelo \LaTeX e posicionamento configurado como `[ht]`.

4.3 Inserindo tabelas e quadros

Na ABNT, tabelas e quadros são tratados de maneira diferente.

- Tabelas: São utilizadas para apresentar informações numéricas de forma organizada, geralmente com linhas e colunas. As tabelas devem ter um título (acima da tabela), uma numeração sequencial e fonte indicativa, se necessário.

Figura 4 – Capa do livro “Métodos Numéricos em Problemas de Engenharia Química” (Pinto; Lage, 2001).



Fonte: Pinto e Lage (2001).

- Quadros: São utilizados para apresentar informações textuais ou descritivas, como dados qualitativos, resumos, descrições, entre outros. Os quadros também devem ter um título (acima do quadro), uma numeração sequencial e fonte indicativa, se necessário.

Ambos elementos devem ser citados e referenciados no texto conforme a ordem em que aparecem, seguindo as normas da ABNT para citações e referências. Além disso, tanto tabelas quanto quadros devem seguir padrões de formatação específicos, como fonte, espaçamento, alinhamento, entre outros, de acordo com as normas da ABNT.

4.3.1 Tabelas

Existem muitas formas de fazer uma tabela. A mais complexa e que vai te tomar um tempo para aprender (coisa que você não deve ter agora) é aprender a linguagem \LaTeX como montar a sua tabela, como pode ser visto na sintaxe abaixo:

```
\begin{table}[htbp]
\centering
\caption{Exemplo de tabela}
\label{tab:exemplo}
```

```

\begin{tabular}{|c|c|}
\hline
\textbf{Coluna 1} & \textbf{Coluna 2} \\
\hline
Dado 1 & Dado 2 \\
Dado 3 & Dado 4 \\
\hline
\end{tabular}
\fonte{Autoria própria.}
\end{table}

```

Neste exemplo, a tabela está dentro do ambiente `table`, que permite adicionar uma legenda e um rótulo para referência cruzada. A opção `[htbp]` especifica as preferências de posicionamento da tabela (aqui, topo, base e em páginas separadas). Além disso:

- Dentro do ambiente `table`, o ambiente `tabular` cria a estrutura da tabela.
- Especificamos o número e a justificação das colunas dentro do comando `\begin{tabular}{...}`, onde `c` significa coluna centralizada.
- Usamos `\hline` para criar linhas horizontais e `&` para separar os elementos de uma mesma linha.
- Após a tabela, usamos `\caption{...}` para adicionar uma legenda à tabela e `\label{...}` para atribuir um rótulo para referência cruzada.

Você pode ajustar o conteúdo e o estilo da tabela conforme necessário para atender aos requisitos do seu documento. O exemplo acima gera a Tabela 1.

Tabela 1 – Exemplo de tabela

Coluna 1	Coluna 2
Dado 1	Dado 2
Dado 3	Dado 4

Fonte: Autoria própria.

Não se preocupe, há métodos mais eficazes disponíveis. Alguns *sites* de qualidade oferecem a capacidade de criar tabelas de forma visual e, em seguida, gerar automaticamente o código \LaTeX correspondente.

Aqui está uma lista de algumas opções, mas lembre-se de que existem outras disponíveis, então recomendo explorar e escolher aquela que mais lhe convém.

- [Clevvert](#)
- [Latex Table Editor](#)
- [Table Generator](#)
- [Table Generator](#)
- [Planilhas do Google](#)

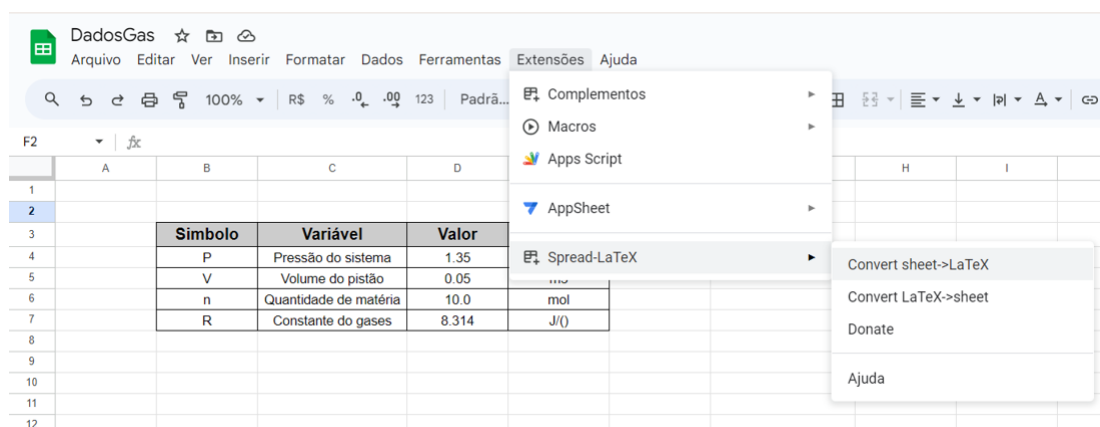
Todos esses geradores irão gerar um código latex que basta você copiar e colar no seu texto. Só três coisas precisam ser verificadas para depois facilitar a sua vida:

1. Editar a legenda da tabela na tag `\caption{}`. Essa tag deve ser colocada ou editada sempre após a tag `\begin{table}`;
2. Adicionar a tag `\label{tabela:nome}` com um código para depois você fazer referência a essa tabela. Adicione essa tag logo após a tag `caption`;
3. Inserir a fonte, antes da tag `\end{table}` usando `\fonte{Do autor}`.

Alguns dos geradores já inserem esses elementos e outros não - você precisa verificar! No mínimo, você vai precisar editar o `caption` (legenda) da figura.

Como exemplo, a Figura 5 mostra como uma tabela criada no "Planilhas do Google" pode ser exportada com a extensão "Spread-LateX" que deve ser instalada por você. O Spread-LateX é um *Add-on* que permite você converter a tabela do google em um código para latex. Uma vez convertida, basta copiar e colar o código no arquivo ".tex".

Figura 5 – Exemplo de como exportar uma tabela do Google para o formato \LaTeX usando a extensão *Spread-LateX*.



Fonte: Autoria própria.

Neste caso, são informados dados de um sistema de gás: pressão do sistema, volume do pistão, quantidade de matéria e constante dos gases, com suas respectivas unidades. Esses dados podem ser vistos na Tabela 2.

Tabela 2 – Tabela criada no Planilhas do Google usando a extensão Spread-LaTeX

Símbolo	Variável	Valor	Unidade
P	Pressão do sistema	1.35	kPa
V	Volume do pistão	0.05	m^3
n	Quantidade de matéria	10.0	mol
R	Constante do gases	8.314	$J/(molK)$

Fonte: Autoria própria.

4.3.2 Quadros

Um quadro não tem uma estrutura muito diferente da tabela. O que vai mudar é que ao invés de usar `\begin{table}` você irá usar o `\begin{quadro}`. Além disso, muitas vezes esse quadro pode ser oriundo de um artigo e você tem ele na forma de imagem. Então ele pode ser adicionado através do seguinte comando:

```
\begin{quadro}
  \caption{Descrição do quadro que você vai adicionar}
  \label{qua:exemplo}
  \includegraphics[scale=0.8]{Arquivos/figuras/Quadro-exemplo.png}
  \fonte{Do Autor.}
\end{quadro}
```

Quadro 1 – Exemplo de quadro usando um figura.

ANO DE FUNDAÇÃO	LOCAL	OBSERVAÇÃO ³
1949	Caxias	Extinta em 1972.
1952	Carolina	Não foi possível o levantar o ano de extinção.
1959	Balsas	Não foi possível o levantar o ano de extinção.
1960	Imperatriz	Extinta em 1979.
1962	Barra do Corda	Não foi possível o levantar o ano de extinção.
1969	Grajaú	Não foi possível o levantar o ano de extinção.
?	Presidente Dutra	Não foi possível o levantar o ano de fundação e nem de extinção, mas sabe-se que a primeira escola (Educandário São Francisco de Assis) foi criada em 1948.
?	Codó	Não foi possível o levantar o ano de fundação e nem de extinção, mas sabe-se que a primeira escola (Santa Filomena) foi criada em 1957 e funciona até os dias de hoje.

Fonte: Autoria própria.

Esse quadros devem ser chamados no texto e explicados de certa forma. Por exemplo, o Quadro 1 mostra como uma figura pode ser inserida com o conteúdo do quadro. Na primeira coluna, pode ser visto que [...] Além disso, o Quadro 2 exemplifica como a estrutura criação de uma tabela pode ser aproveitada para criar quadros. Note que o bem conhecido Método de Newton-Raphson [...]

Quadro 2 – Exemplo de quadro usando uma tabela.

Propriedade	Descrição
Nome do Método	Método de Newton-Raphson
Autor(es)	Isaac Newton e Joseph Raphson
Ano de Publicação	1669
Aplicações	Encontrar raízes de funções

Fonte: Autoria própria.

4.4 Inserindo equações

4.4.1 Equações matemáticas

Inserir equações no \LaTeX é uma tarefa bastante simples e flexível. Ele oferece várias maneiras de criar equações, desde simples expressões matemáticas até equações complexas com múltiplas linhas e símbolos especiais. Além disso, o \LaTeX oferece uma ampla gama de símbolos e comandos matemáticos, bem como pacotes adicionais, como *amsmath*, que fornecem funcionalidades avançadas para manipulação de equações.

A principal ferramenta para isso é o ambiente `equation` para equações numeradas e, para inserir uma equação no meio do texto, você pode usar o comando `$. . . $`. Aqui, estão alguns exemplos de como fazer isso:

- **Inserir uma equação no meio do texto**

Se você quer calcular a velocidade média de um ponto material, você irá calcular através de $v_m = \frac{\Delta x}{\Delta t}$.

Para escrever uma equação no corpo do texto, basta você digitar a equação entre `$. . . $`. A equação acima você foi inserida com o código `$v_m=\frac{\Delta x}{\Delta t}$`.

- **Inserir uma equação numerando-a para posterior referência.**

Por exemplo, definindo a Equação 4.1:

$$v_m = \frac{\Delta x}{\Delta t} \tag{4.1}$$

Essa equação pode ser inserida com o código:

```
\begin{equation}
  \label{eq:velmedia}
  v_m=\frac{\Delta x}{\Delta t}
\end{equation}
```

- **Inserindo uma equação sem numeração.**

Você só vai inserir uma equação sem numeração quando você tiver certeza que não precisará fazer uma referência futura a essa equação. Em geral durante uma dedução, mostramos várias equações que não são referenciadas.

$$v_m = \frac{\Delta x}{\Delta t}$$

```
\begin{equation}
  \nonumber
  v_m=\frac{\Delta x}{\Delta t}
\end{equation}
```

Nesse caso você trocou o `\label{}` que você não vai precisar pelo `\nonumber`.

Já para equações mais complexas, podemos usar o ambiente `align`. Esse ambiente é usado para fazer um empilhamento de equações com a possibilidade de se determinar qual ponto de cada equação deve ser considerado para alinhamento. Como todo ambiente, o `align` deve ser iniciado e fechado com os comandos `\begin{align}` e `\end{align}`.

$$f(x) = x^2 \tag{4.2}$$

$$g(x) = \frac{1}{x} \tag{4.3}$$

$$F(x) = \int_a^b f(x) dx \tag{4.4}$$

Seguem outros exemplos de equações:

- Equação com raiz quadrada:

$$f(x) = \sqrt{\frac{2x^2 + 3}{x + 1}} \tag{4.5}$$

- Equação com frações e índices:

$$\frac{\partial^2 u}{\partial t^2} = c^2 \nabla^2 u \tag{4.6}$$

- Equação com somatório:

$$\sum_{i=1}^n x_i^2 \tag{4.7}$$

- Equação diferencial parcial:

$$\frac{\partial^2 u}{\partial t^2} = c^2 \frac{\partial^2 u}{\partial x^2} \quad (4.8)$$

- Equação com matriz:

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \quad (4.9)$$

Finalmente, aqui está um exemplo de uma equação diferencial parcial que combina efeitos advectivos na direção axial z e efeitos difusivos na direção radial r :

$$\frac{\partial C}{\partial t} = D_r \left(\frac{\partial^2 C}{\partial r^2} + \frac{1}{r} \frac{\partial C}{\partial r} \right) - v \frac{\partial C}{\partial z} \quad (4.10)$$

As Equações (4.11a) a (4.11d) descrevem as condições iniciais e de contorno.

$$C(r, z, t = 0) = C_0(r, z) \quad (4.11a)$$

$$C(r = R, z, t) = C_{ref}(z, t) \quad (4.11b)$$

$$\left. \frac{\partial C}{\partial r} \right|_{r=0} = 0 \quad (4.11c)$$

$$C(r, z = 0, t) = C_{feed}(r, t) \quad (4.11d)$$

Na Equação 4.10,

- C é a concentração de uma substância,
- t é o tempo,
- D_r é o coeficiente de difusão na direção radial,
- v é a velocidade na direção axial.

4.4.2 Equações químicas

Os pacotes `chemfig` e `mchem` são amplamente utilizados para representar fórmulas químicas de maneira elegante e profissional no \LaTeX . Ambos os pacotes são indispensáveis para quem trabalha com documentos que envolvem química, seja em relatórios de laboratório, artigos científicos ou materiais didáticos. Sua facilidade de uso e qualidade de saída tornam a representação de fórmulas químicas uma tarefa simples e agradável.

4.4.2.1 Usando o pacote `chemfig`

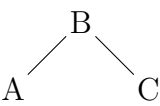
O pacote `chemfig` oferece uma ampla gama de ferramentas para desenhar estruturas químicas, incluindo moléculas, reações químicas e diagramas. Ele permite a construção detalhada de fórmulas com facilidade, usando uma sintaxe intuitiva e flexível.

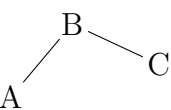
Escrever fórmulas químicas com o *chemfig* é direto. Por exemplo, `\chemfig{A=B}` desenha a molécula. O símbolo `=` determina o tipo de ligação:

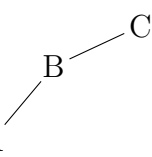
- Ligação simples: O — H
- Ligação dupla: O = H
- Ligação tripla: O ≡ H
- Ligação à esquerda: O ◀ H

Consulte o guia de referência para obter uma lista de tipos de ligação.

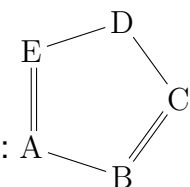
É possível definir uma fórmula química usando unidades que definem os ângulos entre as ligações. Usando o comando `\chemfig{A-[1]B-[7]C}`, os parâmetros dentro dos colchetes definem o ângulo em unidades especiais, onde cada unidade corresponde a 45°. Portanto, no exemplo, os ângulos são 45° e 315°. Podem ser definidos:

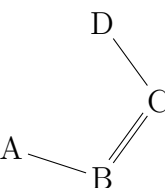
Ângulos na unidade padrão: 

Ângulos absolutos: 

Ângulos relativos: 

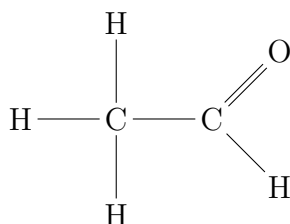
Podem-se escrever estruturas com anéis, sendo:

polígonos regulares: 

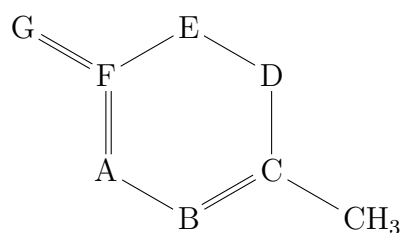
Polígonos irregulares também são possíveis: 

As fórmulas químicas nem sempre são lineares; na verdade, as fórmulas ramificadas são o tipo mais comum. Abaixo está um exemplo de como criá-las:

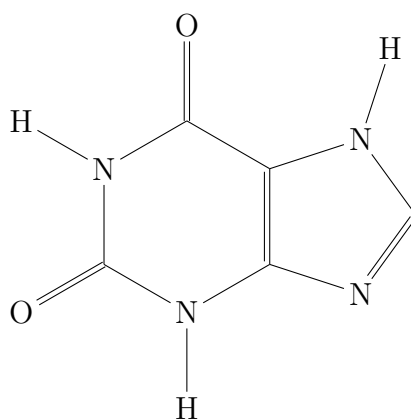
- Ramificação: $\text{\chemfig{H-C(-[2]H)(-[6]H)-C(=[1]O)-[7]H}}$



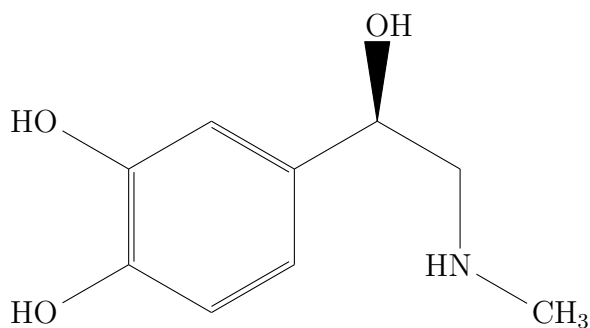
- Anel com ramificação: $\text{\chemfig{A*6(-B=C(-CH_3)-D-E-F(=G)=)}}$



Com isso, moléculas orgânicas podem ser facilmente criadas. Veja o exemplo da cafeína, com o comando $\text{\chemfig{*6((=O)-N(-H)-(*5(-N=-N(-H)-))=-(=O)-N(-H)-)}}$:

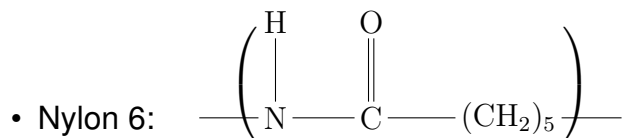
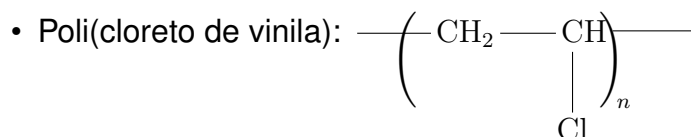


Outro exemplo é a adrenalina:



Representações de unidades básicas de polímeros também pode ser construídas:

- Polietileno: $\text{---(CH}_2\text{---CH}_2\text{)}_n\text{---}$



Veja mais detalhes no Anexo [B](#)

4.4.2.2 Usando o pacote mhchem

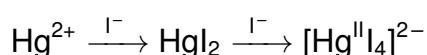
Dando continuidade a esta seção, o pacote `mhchem` é especialmente útil para escrever equações químicas de forma rápida e eficiente. Ele fornece comandos simples para inserir reações químicas balanceadas, além de suportar a formatação de elementos químicos e símbolos comuns. Seguem básicos usando o pacote `mhchem`:

Moléculas simples:

- Water: `\ce{H2O}`, H_2O
- Benzene: `\ce{C6H6}`, C_6H_6
- Hydrogen peroxide: `\ce{H2O2}`, H_2O_2
- Acetic acid: `\ce{C2H4O2}`, $\text{C}_2\text{H}_4\text{O}_2$
- Glucose: `\ce{C6H12O6}`, $\text{C}_6\text{H}_{12}\text{O}_6$

Reações químicas:

- `\ce{2H2 + O2 -> 2H2O}` representa a reação $2\text{H}_2 + \text{O}_2 \longrightarrow 2\text{H}_2\text{O}$
- `\ce{CO2 + C -> 2 CO}` representa a reação $\text{CO}_2 + \text{C} \longrightarrow 2\text{CO}$
- `\ce{Hg^{2+} ->[I-] HgI2 ->[I-] [Hg^{II}I4]^{2-}}` representa a reação



As expressões químicas podem ser formatadas usando comandos do modo matemático. Com a sintaxe de `\frac`, é possível escrever a expressão:

$$K = \frac{[\text{Hg}^{2+}][\text{Hg}]}{[\text{Hg}_2^{2+}]}$$

4.5 Notas de Rodapé

Por fim, as notas de rodapé ¹ A inserção de uma nota de rodapé é feita colocando `\footnote{Escreva aqui seu texto}` exatamente onde você quer fazer menção a nota de rodapé.

¹ Notas de rodapé são explicações ou comentários que são colocados no fim da página e que não cabe ser colocada no corpo do texto mas pode ser importante para alguns leitores.

5 MATERIAIS E MÉTODOS

Neste capítulo, serão apresentados os materiais utilizados e os métodos empregados na realização deste estudo. Descreveremos detalhadamente os procedimentos experimentais e as ferramentas utilizadas para coletar e analisar os dados necessários para atingir os objetivos propostos.

A seleção adequada dos materiais e a aplicação de métodos precisos são fundamentais para garantir a confiabilidade e a precisão dos resultados obtidos. Assim, este capítulo fornecerá uma visão geral das etapas metodológicas adotadas, desde a coleta de amostras até a análise estatística dos dados.

Além disso, destacaremos eventuais modificações ou adaptações realizadas nos métodos padrão, bem como as justificativas por trás dessas decisões. Isso proporcionará uma compreensão mais abrangente do processo experimental e permitirá a replicabilidade do estudo por outros pesquisadores interessados.

Por fim, será enfatizada a importância da rigorosidade metodológica na condução da pesquisa científica, visando garantir a validade e a relevância dos resultados obtidos.

6 RESULTADOS

Um Trabalho de Conclusão de Curso (TCC) é uma peça acadêmica com o objetivo principal de demonstrar a capacidade do aluno de aplicar os conhecimentos adquiridos durante o curso para investigar, analisar e resolver um problema ou questão específica dentro de sua área de estudo. Os seguintes pontos podem ser relacionados como resultados esperados em um TCC:

1. Identificação e análise crítica de um problema ou questão relevante dentro da área de estudo.
2. Aplicação de métodos de pesquisa apropriados para coletar e analisar dados relacionados ao problema investigado.
3. Desenvolvimento de argumentos convincentes com base em evidências sólidas e análise crítica da literatura relevante.
4. Apresentação clara e organizada dos resultados obtidos, incluindo tabelas, gráficos e outras representações visuais, quando apropriado.
5. Discussão aprofundada dos resultados, incluindo interpretação dos achados, implicações teóricas e práticas, e sugestões para pesquisas futuras.
6. Conclusões bem fundamentadas que resumam os principais resultados e contribuições do trabalho, destacando suas limitações e áreas para futuras investigações.
7. Demonstração de habilidades de escrita acadêmica, incluindo uso correto de citações e referências de acordo com as normas da área e da instituição.
8. Apresentação oral eficaz do trabalho, incluindo capacidade de comunicar claramente os objetivos, métodos, resultados e conclusões.
9. Contribuição para o avanço do conhecimento na área de estudo, seja por meio da resolução de um problema específico, da confirmação ou refutação de hipóteses, ou da proposição de novas abordagens ou perspectivas.

Estes são alguns dos resultados esperados em um TCC, que refletem o objetivo geral de demonstrar a capacidade do aluno de realizar uma investigação independente e contribuir para o corpo de conhecimento em sua área de estudo.

7 CRONOGRAMA

Criar um cronograma eficaz para o TCC é essencial para garantir que todas as etapas do projeto sejam concluídas dentro do prazo. Aqui estão algumas dicas para elaborar um cronograma:

1. **Divida o projeto em etapas:** Identifique todas as tarefas necessárias para concluir o TCC, desde a revisão da literatura até a defesa oral. Divida essas tarefas em etapas menores e mais gerenciáveis.
2. **Estime o tempo necessário para cada etapa:** Determine quanto tempo será necessário para concluir cada etapa do projeto. Considere fatores como a complexidade da tarefa, disponibilidade de recursos e experiência prévia.
3. **Estabeleça prazos realistas:** Defina prazos realistas para cada etapa do projeto, levando em consideração a quantidade de trabalho envolvido e outros compromissos pessoais ou acadêmicos.
4. **Priorize as tarefas:** Identifique as tarefas mais críticas e atribua prioridades a elas. Certifique-se de reservar tempo suficiente para as etapas mais importantes e que demandam mais tempo.
5. **Revise e ajuste o cronograma regularmente:** À medida que o projeto avança, revise e ajuste o cronograma conforme necessário. Esteja preparado para lidar com imprevistos e atrasos, e faça alterações no cronograma conforme apropriado.
6. **Use ferramentas de gerenciamento de projetos:** Considere o uso de ferramentas de gerenciamento de projetos, como planilhas, aplicativos ou softwares especializados, para ajudar a monitorar e acompanhar o progresso do cronograma.
7. **Mantenha-se flexível:** Seja flexível e esteja aberto a ajustar o cronograma conforme necessário. Às vezes, certas etapas podem levar mais tempo do que o esperado, e é importante ser capaz de adaptar o cronograma conforme surgem desafios.
8. **Estabeleça marcos importantes:** Identifique marcos importantes no cronograma, como a conclusão da revisão da literatura, a submissão do projeto escrito e a defesa oral. Isso ajudará a manter o projeto no caminho certo e a acompanhar o progresso.

REFERÊNCIAS

- BAPTESTINI, F. M.; CORRÊA, P. C.; JUNQUEIRA, M. S.; RAMOS, A. M.; VANEGAS, J. D.; COSTA, C. F. Mathematical modeling of drying of soursop foam. *Revista Brasileira de Engenharia Agrícola e Ambiental*, Departamento de Engenharia Agrícola - UFCG/Cnpq, v. 19, p. 1203–1208, 12 2015. ISSN 18071929. Citado na página 20.
- DELLAMATRICE, P. M.; MONTEIRO, R. T. R. Principais aspectos da poluição de rios brasileiros por pesticidas. *Revista Brasileira de Engenharia Agrícola e Ambiental*, scielo, v. 18, p. 1296–1301, 12 2014. Disponível em: <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S1415-43662014001200014&lang=pt>. Citado na página 19.
- GUIZZARDI, G. *Ontological Foundations for Structural Conceptual Models*. Tese (Doutorado) — Centre for Telematics and Information Technology, University of Twente, Enschede, The Netherlands, 2005. Disponível em: <<http://www.loa.istc.cnr.it/Guizzardi/SELMAS-CR.pdf>>. Acesso em: 3 jul. 2011. Citado na página 20.
- MCDONAGH, J. L.; ZAVITSANOU, S.; HARRISON, A.; ZUBAREV, D.; van Kessel, T.; WUNSCH, B. H.; CIPCIGAN, F. Chemical space analysis and property prediction for carbon capture solvent molecules††electronic supplementary information (esi) available. see doi: <https://doi.org/10.1039/d3dd00073g>. *Digital Discovery*, v. 3, n. 3, p. 528–543, 2024. ISSN 2635-098X. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S2635098X2400041X>>. Citado 2 vezes nas páginas 18 e 20.
- MINASIDIS, V.; JÄSCHKE, J.; SKOGESTAD, S. Economic plantwide control: Automated controlled variable selection for a reactor-separator-recycle process. *IFAC Proceedings Volumes*, v. 46, n. 32, p. 87–92, 2013. ISSN 1474-6670. 10th IFAC International Symposium on Dynamics and Control of Process Systems. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S1474667015382392>>. Citado 3 vezes nas páginas 18, 20 e 23.
- PINTO, J. C.; LAGE, P. L. da C. *Métodos numéricos em problemas de engenharia química*. Rio de Janeiro: Editora E-papers, 2001. Citado 3 vezes nas páginas 7, 22 e 26.
- SAMPAIO, C. *Visualizando informações com um gráfico*. 2011. Acessado em: 1 jun. 2011. Disponível em: <<https://www.alura.com.br/artigos/visualizando-informacoes-com-um-grafico>>. Citado na página 24.
- SCHWAAB, M.; BISCAIA, E. C.; MONTEIRO, J. L.; PINTO, J. C. Nonlinear parameter estimation through particle swarm optimization. *Chemical Engineering Science*, Pergamon, v. 63, p. 1542–1552, 3 2008. ISSN 0009-2509. Citado 2 vezes nas páginas 18 e 20.
- van GIGCH, J. P.; PIPINO, L. L. In search for a paradigm for the discipline of information systems. *Future Computing Systems*, v. 1, n. 1, p. 71–97, 1986. Citado na página 18.
- WILSON, P.; MADSEN, L. *The Memoir Class for Configurable Typesetting - User Guide*. Normandy Park, WA, 2010. Disponível em: <<http://mirrors.ctan.org/macros/latex/>>

[contrib/memoir/memman.pdf](#)>. Acesso em: 19 dez. 2012. Citado 2 vezes nas páginas 18 e 20.

APÊNDICE A – LOGOMARCA DE UFES

A Figura 6 faz parte da Identidade Visual da Universidade Federal do Espírito Santo (Ufes), que tem por objetivo padronizar documentos, fortalecer e divulgar a marca institucional. A Identidade Visual é uma produção da Superintendência de Comunicação da Ufes.

Figura 6 – Logomarca da UFES.



Fonte: [Manual de Identidade Visual da UFES](#).

**APÊNDICE B – NULLAM ELEMENTUM URNA VEL IMPERDIET SODALES ELIT
IPSUM PHARETRA LIGULA AC PRETIUM ANTE JUSTO A NULLA CURABITUR
TRISTIQUE ARCU EU METUS**

Nunc velit. Nullam elit sapien, eleifend eu, commodo nec, semper sit amet, elit. Nulla lectus risus, condimentum ut, laoreet eget, viverra nec, odio. Proin lobortis. Curabitur dictum arcu vel wisi. Cras id nulla venenatis tortor congue ultrices. Pellentesque eget pede. Sed eleifend sagittis elit. Nam sed tellus sit amet lectus ullamcorper tristique. Mauris enim sem, tristique eu, accumsan at, scelerisque vulputate, neque. Quisque lacus. Donec et ipsum sit amet elit nonummy aliquet. Sed viverra nisl at sem. Nam diam. Mauris ut dolor. Curabitur ornare tortor cursus velit.

Morbi tincidunt posuere arcu. Cras venenatis est vitae dolor. Vivamus scelerisque semper mi. Donec ipsum arcu, consequat scelerisque, viverra id, dictum at, metus. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut pede sem, tempus ut, porttitor bibendum, molestie eu, elit. Suspendisse potenti. Sed id lectus sit amet purus faucibus vehicula. Praesent sed sem non dui pharetra interdum. Nam viverra ultrices magna.

Aenean laoreet aliquam orci. Nunc interdum elementum urna. Quisque erat. Nullam tempor neque. Maecenas velit nibh, scelerisque a, consequat ut, viverra in, enim. Duis magna. Donec odio neque, tristique et, tincidunt eu, rhoncus ac, nunc. Mauris malesuada malesuada elit. Etiam lacus mauris, pretium vel, blandit in, ultricies id, libero. Phasellus bibendum erat ut diam. In congue imperdiet lectus.

ANEXO A – REFERÊNCIAS RÁPIDAS EM \LaTeX

L^AT_EX 2_ε Cheat Sheet

Document classes

`book` Default is two-sided.
`report` No `\part` divisions.
`article` No `\part` or `\chapter` divisions.
`letter` Letter (?).
`slides` Large sans-serif font.

Used at the very beginning of a document: `\documentclass{class}`.
Use `\begin{document}` to start contents and `\end{document}` to end the document.

Common documentclass options

`10pt/11pt/12pt` Font size.
`letterpaper/a4paper` Paper size.
`wocolumn` Use two columns.
`twoside` Set margins for two-sided.
`landscape` Landscape orientation. Must use `dvips -t landscape`.
`draft` Double-space lines.
Usage: `\documentclass[opt, opt]{class}`.

Packages

`fullpage` Use 1 inch margins.
`anysize` Set margins: `\marginwidth{l}{r}{t}{b}`.
`multicol` Use *n* columns: `\begin{multicols}{n}`.
`latexsym` Use L^AT_EX symbol font.
`graphicx` Show image: `\includegraphics[width=x]{file}`.
`url` Insert URL: `\url{http://...}`.
Use before `\begin{document}`. Usage: `\usepackage{package}`

Title

`\author{text}` Author of document.
`\title{text}` Title of document.
`\date{text}` Date.
These commands go before `\begin{document}`. The declaration `\maketitle` goes at the top of the document.

Miscellaneous

`\pagestyle{empty}` Empty header, footer and no page numbers.
`\tableofcontents` Add a table of contents here.

Document structure

`\part{title}` `\subsection{title}`
`\chapter{title}` `\paragraph{title}`
`\section{title}` `\subparagraph{title}`
`\subsection{title}`

Use `\setcounter{secnumdepth}{x}` suppresses heading numbers of depth $> x$, where `chapter` has depth 0. Use a `*`, as in `\section*{title}`, to not number a particular item—these items will also not appear in the table of contents.

Text environments

`\begin{comment}` Comment (not printed). Requires `verbatim` package.
`\begin{quote}` Indented quotation block.
`\begin{quotation}` Like `quote` with indented paragraphs.
`\begin{verse}` Quotation block for verse.

Lists

`\begin{enumerate}` Numbered list.
`\begin{itemize}` Bulleted list.
`\begin{description}` Description list.
`\item text` Add an item.
`\item[x] text` Use *x* instead of normal bullet or number. Required for descriptions.

References

`\label{marker}` Set a marker for cross-reference, often of the form `\label{sec: item}`.
`\ref{marker}` Give section/body number of marker.
`\pageref{marker}` Give page number of marker.
`\footnote{text}` Print footnote at bottom of page.

Floating bodies

`\begin{table}[place]` Add numbered table.
`\begin{figure}[place]` Add numbered figure.
`\begin{equation}[place]` Add numbered equation.
`\caption{text}` Caption for the body.
The *place* is a list valid placements for the body. `t=top`, `b=bottom`, `p=separate page`, `!=place even if ugly`. Captions and label markers should be within the environment.

Text properties

Font face

Command	Declaration	Effect
<code>\textrm{<i>text</i>}</code>	<code>\rmfamily <i>text</i></code>	Roman family
<code>\textsf{<i>text</i>}</code>	<code>\sffamily <i>text</i></code>	Sans serif family
<code>\texttt{<i>text</i>}</code>	<code>\ttfamily <i>text</i></code>	Typewriter family
<code>\textmd{<i>text</i>}</code>	<code>\mdseries <i>text</i></code>	Medium series
<code>\textbf{<i>text</i>}</code>	<code>\bfseries <i>text</i></code>	Bold series
<code>\textup{<i>text</i>}</code>	<code>\upshape <i>text</i></code>	Upright shape
<code>\textit{<i>text</i>}</code>	<code>\itshape <i>text</i></code>	<i>Italic shape</i>
<code>\textsl{<i>text</i>}</code>	<code>\slshape <i>text</i></code>	<i>Slanted shape</i>
<code>\textsc{<i>text</i>}</code>	<code>\scshape <i>text</i></code>	SMALL CAPS SHAPE
<code>\emph{<i>text</i>}</code>	<code>\em <i>text</i></code>	<i>Emphasized</i>
<code>\textnormal{<i>text</i>}</code>	<code>\normalfont <i>text</i></code>	Document font
<code>\underline{<i>text</i>}</code>		<u>Underline</u>

The command `(tttt)` form handles spacing better than the declaration `(tttt)` form.

Font size

<code>\tiny</code>	<code>tiny</code>	<code>\Large</code>	Large
<code>\scriptsize</code>	<code>scriptsize</code>	<code>\LARGE</code>	LARGE
<code>\footnotesize</code>	<code>footnotesize</code>	<code>\huge</code>	huge
<code>\small</code>	<code>small</code>	<code>\Huge</code>	Huge
<code>\normalsize</code>	<code>normalsize</code>		
<code>\large</code>	<code>large</code>		

These are declarations and should be used in the form `{\small ...}`, or without braces to affect the entire document.

Verbatim text

`\begin{verbatim}` Verbatim environment.
`\begin{verbatim*}` Spaces are shown as `␣`.
`\verb!text!` Text between the delimiting characters (in this case '!') is verbatim.

Justification

Environment	Declaration
<code>\begin{center}</code>	<code>\centering</code>
<code>\begin{flushleft}</code>	<code>\raggedright</code>
<code>\begin{flushright}</code>	<code>\raggedleft</code>

Miscellaneous

`\linespread{x}` changes the line spacing by the multiplier *x*.

Text-mode symbols

Symbols

<code>&</code>	<code>\&</code>	<code>-</code>	<code>_</code>	<code>...</code>	<code>\ldots</code>	<code>•</code>	<code>\textbullet</code>
<code>\$</code>	<code>\\$</code>	<code>~</code>	<code>\~{}</code>	<code> </code>	<code>\textbar</code>	<code>\</code>	<code>\textbackslash</code>
<code>%</code>	<code>\%</code>	<code>#</code>	<code>\#</code>	<code>#</code>	<code>\#</code>	<code>§</code>	<code>\S</code>

Accents

<code>ò</code>	<code>\'o</code>	<code>ó</code>	<code>\'o</code>	<code>ô</code>	<code>\^o</code>	<code>õ</code>	<code>\~o</code>	<code>ö</code>	<code>\=o</code>
<code>ö</code>	<code>\.o</code>	<code>ö</code>	<code>\"o</code>	<code>q</code>	<code>\c o</code>	<code>ö</code>	<code>\v o</code>	<code>ö</code>	<code>\H o</code>
<code>ç</code>	<code>\c c</code>	<code>ç</code>	<code>\d o</code>	<code>ç</code>	<code>\b o</code>	<code>ö</code>	<code>\t oo</code>	<code>œ</code>	<code>\oe</code>
<code>Æ</code>	<code>\OE</code>	<code>æ</code>	<code>\ae</code>	<code>Æ</code>	<code>\AE</code>	<code>ä</code>	<code>\aa</code>	<code>Å</code>	<code>\AA</code>
<code>ø</code>	<code>\o</code>	<code>Ø</code>	<code>\O</code>	<code>ı</code>	<code>\l</code>	<code>ı</code>	<code>\L</code>	<code>ı</code>	<code>\i</code>
<code>ı</code>	<code>\j</code>	<code>ı</code>	<code>\j</code>	<code>ı</code>	<code>\j</code>	<code>ı</code>	<code>\j</code>	<code>ı</code>	<code>\j</code>

Delimiters

<code>'</code>	<code>"</code>	<code>{</code>	<code>[</code>	<code>(</code>	<code><</code>	<code>\textless</code>
<code>'</code>	<code>"</code>	<code>}</code>	<code>]</code>	<code>)</code>	<code>></code>	<code>\textgreater</code>

Dashes

Name	Source	Example	Usage
<code>hyphen</code>	<code>-</code>	X-ray	In words.
<code>en-dash</code>	<code>--</code>	1-5	Between numbers.
<code>em-dash</code>	<code>---</code>	Yes—or no?	Punctuation.

Line and page breaks

`\` Begin new line without new paragraph.
`*` Prohibit pagebreak after linebreak.
`\kill` Don't print current line.
`\pagebreak` Start new page.
`\noindent` Do not indent current line.

Miscellaneous

`\today` March 28, 2017.
`\sim` Prints `~` instead of `\~{}`, which makes `~`.
`-` Space, disallow linebreak (W.J.'Clinton).
`\.` Indicate that the `.` ends a sentence when following an uppercase letter.
`\hspace{l}` Horizontal space of length *l* (Ex: *l* = 20pt).
`\vspace{l}` Vertical space of length *l*.
`\rule{w}{h}` Line of width *w* and height *h*.

Tabular environments

tabbing environment

`\=` Set tab stop. `\>` Go to tab stop.
Tab stops can be set on "invisible" lines with `\kill` at the end of the line. Normally `\` is used to separate lines.

tabular environment

```
\begin{array}[pos]{cols}
\begin{tabular}[pos]{cols}
\begin{tabular*}[width][pos]{cols}
```

tabular column specification

l Left-justified column.
c Centered column.
r Right-justified column.
p{width} Same as \parbox[t]{width}.
{decl} Insert decl instead of inter-column space.
| Inserts a vertical line between columns.

tabular elements

\hline Horizontal line between rows.
\cline{x-y} Horizontal line across columns x through y.
\multicolumn{n}{cols}{text} A cell that spans n columns, with cols column specification.

Math mode

For inline math, use \dots or $\$...\$$. For displayed math, use $\[...\]$ or $\begin{equation}$.

Superscript^x \hat{x} Subscript_x $_x$
 $\frac{x}{y}$ $\frac{x}{y}$ $\sum_{k=1}^n$ $\sum_{k=1}^n$
 $\sqrt[n]{x}$ $\sqrt[n]{x}$ $\prod_{k=1}^n$ $\prod_{k=1}^n$

Math-mode symbols

\leq \leq \geq \geq \neq \neq \approx \approx
 \times \times \div \div \pm \pm \cdot \cdot
 \circ \circ \sim \sim \prime \prime \dots \dots
 ∞ \infty \neg \neg \wedge \wedge \vee \vee
 \supset \supset \forall \forall \in \in \rightarrow \rightarrow
 \subset \subset \exists \exists \notin \notin \Rightarrow \Rightarrow
 \cup \cup \cap \cap $|$ \mid \Leftrightarrow \Leftrightarrow
 \dot{a} \dot{a} \hat{a} \hat{a} \bar{a} \bar{a} \tilde{a} \tilde{a}
 α \alpha β \beta γ \gamma δ \delta
 ϵ \epsilon ζ \zeta η \eta ε \varepsilon
 θ \theta ι \iota κ \kappa ϑ \vartheta
 λ \lambda μ \mu ν \nu ξ \xi
 π \pi ρ \rho σ \sigma τ \tau
 υ \upsilon ϕ \phi χ \chi ψ \psi
 ω \omega Γ \Gamma Δ \Delta Θ \Theta
 Λ \Lambda Ξ \Xi Π \Pi Σ \Sigma
 Υ \Upsilon Φ \Phi Ψ \Psi Ω \Omega

Bibliography and citations

When using BibTeX, you need to run latex, bibtex, and latex twice more to resolve dependencies.

Citation types

\cite{key} Full author list and year. (Watson and Crick 1953)
\citeA{key} Full author list. (Watson and Crick)
\citeN{key} Full author list and year. Watson and Crick (1953)
\shortcite{key} Abbreviated author list and year. ?
\shortciteA{key} Abbreviated author list. ?
\shortciteN{key} Abbreviated author list and year. ?
\citeyear{key} Cite year only. (1953)
All the above have an NP variant without parentheses; Ex. \citeNP.

BibTeX entry types

@article Journal or magazine article.
@book Book with publisher.
@booklet Book without publisher.
@conference Article in conference proceedings.
@inbook A part of a book and/or range of pages.
@incollection A part of book with its own title.
@misc If nothing else fits.
@phdthesis PhD. thesis.
@proceedings Proceedings of a conference.
@techreport Tech report, usually numbered in series.
@unpublished Unpublished.

BibTeX fields

address Address of publisher. Not necessary for major publishers.
author Names of authors, of format
booktitle Title of book when part of it is cited.
chapter Chapter or section number.
edition Edition of a book.
editor Names of editors.
institution Sponsoring institution of tech. report.
journal Journal name.
key Used for cross ref. when no author.
month Month published. Use 3-letter abbreviation.
note Any additional information.
number Number of journal or magazine.
organization Organization that sponsors a conference.
pages Page range (2,6,9--12).
publisher Publisher's name.
school Name of school (for thesis).
series Name of series of books.
title Title of work.
type Type of tech. report, ex. "Research Note".
volume Volume of a journal or book.
year Year of publication.
Not all fields need to be filled. See example below.

Common BibTeX style files

abbrv Standard abstract alpha with abstract
alpha Standard apa APA
plain Standard unsrt Unsorted

The L^AT_EX document should have the following two lines just before \end{document}, where bibfile.bib is the name of the BibTeX file.

```
\bibliographystyle{plain}
\bibliography{bibfile}
```

BibTeX example

The BibTeX database goes in a file called file.bib, which is processed with bibtex file.

```
@String{N = {Na\-ture}}
@Article{WC:1953,
  author = {James Watson and Francis Crick},
  title = {A structure for Deoxyribose Nucleic Acid},
  journal = N,
  volume = {171},
  pages = {737},
  year = 1953
}
```

Sample L^AT_EX document

```
\documentclass[11pt]{article}
\usepackage{fullpage}
\title{Template}
\author{Name}
\begin{document}
\maketitle

\section{section}
\subsection*{subsection without number}
text \textbf{bold text} text. Some math:  $\$2+2=5\$$ 
\subsection{subsection}
text \emph{emphasized text} text. \cite{WC:1953}
discovered the structure of DNA.
```

```
A table:
\begin{table}[!th]
\begin{tabular}{|l|c|r|}
\hline
first & row & data \\
second & row & data \\
\hline
\end{tabular}
\caption{This is the caption}
\label{ex:table}
\end{table}
```

The table is numbered \ref{ex:table}.
\end{document}

Copyright © 2014 Winston Chang
<http://wch.github.io/latexsheet/>

ANEXO B – CHEMFIG: UM PACOTE PARA DESENHAR MOLÉCULAS L^AT_EX

chemfig

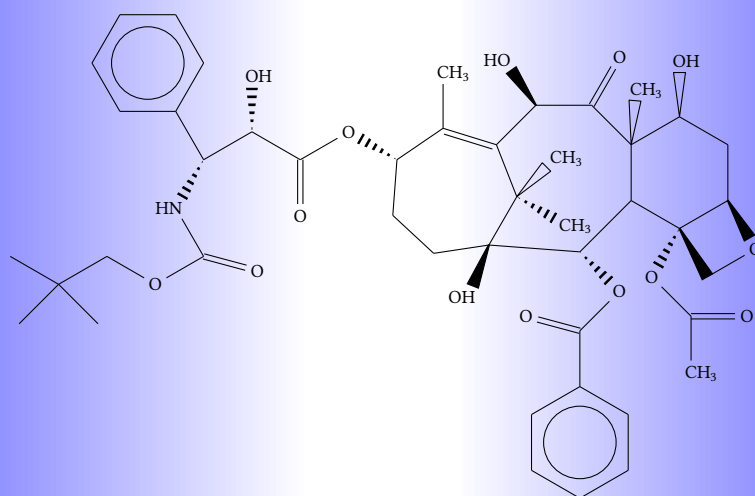
v1.66

28 december 2023

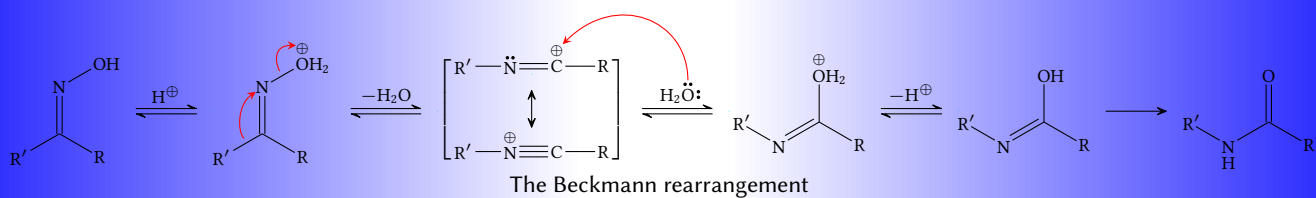
Christian Tellechea

unbonpetit@netc.fr

A T_EX package for drawing molecules



Taxotere



Contents

I	Introduction	4
1	New in v1.6	4
1.1	Lewis decorations	4
1.2	Key debug	4
1.3	Token # in schemes	4
1.4	Key gchemname	4
1.5	Keys "schemestart code" and "schemestop code"	4
1.6	Key "baseline"	4
1.7	Cram bonds join	5
2	Presenting chemfig	5
3	Acknowledgment	5
II	Operation of chemfig	6
1	The \chemfig macro	6
2	Groups of atoms	7
3	First atom's role	7
4	Different types of bonds	7
5	Bond angle	9
5.1	Predefined angles	9
5.2	Absolute angles	10
5.3	Relative angles	10
6	Length of a bond	10
7	Departure and arrival atoms	11
8	Customization of bonds	12
9	Connecting bonds	12
10	Default values	13
11	Branches	14
11.1	Principle	14
11.2	Nesting	15
11.3	Method	15
12	Connecting distant atoms	15
13	Rings	17
13.1	Syntax	17
13.2	Angular position	18
13.2.1	At the start	18
13.2.2	After a bond	18
13.3	Branches on a ring	19
13.4	Nested rings	20
13.5	Rings and groups of atoms	20
13.6	Center of rings	21

14 Representing electron movements	21
14.1 Mesomeric effects	22
14.2 Reaction mechanisms	23
15 Writing a name under a molecule	24
III Advanced usage	26
1 Separating atoms	26
2 Displaying atoms	26
3 Arguments given to tikz	27
4 Shifted double bonds	27
5 Delocalized double bonds	28
6 Saving a sub-molecule	28
7 Placement of Atoms	30
7.1 Groups of atoms	30
7.2 Vertical alignment	30
7.3 Bonds between atoms	31
7.4 La macro <code>\chemskipalign</code>	33
8 The macro <code>\charge</code>	33
8.1 Overview	33
8.2 Parameters	34
8.3 Lewis formula	35
8.4 Integration in <code>chemfig</code>	35
9 Stacking	36
10 Using <code>\chemfig</code> in the <code>tikzpicture</code> environment	36
11 Annotated examples	37
11.1 Ethanal	37
11.2 2-amino-4-oxohexanoic acid	37
11.2.1 Absolute angles	38
11.2.2 Relative angles	38
11.2.3 Ring	38
11.2.4 Nested rings	39
11.3 Glucose	39
11.3.1 Skeleton diagram	39
11.3.2 Fisher projection	40
11.3.3 “Chair” representation	41
11.3.4 Haworth projection	41
11.4 Adrenaline	42
11.4.1 Using one ring	42
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 Draw a polymer element	46
12.5 Draw the symmetrical of a molecule	47
12.6 Add text above bonds and arc to angles	48

12.7 Dessiner des liaisons multiples	49
IV Reaction schemes	49
1 Overview	49
2 Arrow types	50
3 Arrows features	50
4 Compounds names	51
5 Anchoring	52
6 Compounds style	53
7 Branching	54
8 Subscheme	55
9 Arrows optional arguments	57
10 Arrows customization	59
10.1 First arrow	60
10.2 Curved arrow	61
11 The \merge command	62
12 The + sign	64
V List of commands	66
VI Gallery	68

Introduction

1 New in v1.6

1.1 Lewis decorations

As announced since version 1.5 of 5/3/2020, the deprecated macros `lewis` and `Lewis` are no longer available in the `chemfig` package. The recommended method for drawing Lewis formulas is to use `\charge` and `\Charge`, see page 33.

If the use of the macros `lewis` and `Lewis` is indispensable, their code is in the file `chemfig-lewis.tex` which it is possible to load using `input`, after having loaded the package `chemfig`.

1.2 Key `debug`

A new boolean key `debug`, false by default, is available. When set to true, the (rectangular) outline of each group of atoms is drawn in red and the outline of each atom is drawn in gray. The number of the group of atoms is shown above the red rectangle, and similarly for the number of each atom (the atoms are numbered from left to right, starting from 1). These numbers allow to know the name of the node of each atom whose syntax is `n<a>-` where `<a>` is the number of the atom group and `` is the number of the atom.

In this example, a blue arrow starts at atom `n1-3`, which is "C2", and goes to atom `n2-4`, which is "Gz".

Name of nodes
<pre>\setchemfig{debug=true} \chemfig{A1BC2-[:30]DxEyFGz-H3I} \chemmove{\draw[blue](n1-3)to[out=75,in=90](n2-4);}</pre>

1.3 Token # in schemes

In a scheme, the `#` token is now allowed when in the argument of the `\chemfig` macro. See page 8.

1.4 Key `gchemname`

This key, true by default, makes the maximum depth assignments made by `\chemnameinit` and `\chemname` global or local.

1.5 Keys "schemestart code" and "schemestop code"

The values (empty by default) contained in these two keys are executed at the very beginning and at the very end of a reaction scheme. They are not executed in nested reaction schemes.

For example, we can write `schemestart code=\chemnameint{}` to reset the name depth at the beginning of each reaction scheme.

1.6 Key "baseline"

The value passed to this key sets the vertical position of the molecule. See page 30.

1.7 Cram bonds join

When `bond join` is `<true>`, Cram bonds better connect to each other and to simple bonds. See page 13.

2 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 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$;

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

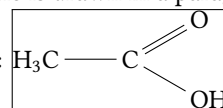
The most important command for drawing molecules is `\chemfig{<code>}`. The argument `code` is a set of characters describing the structure of the molecule according to the rules which are described in this manual.

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: `pdf ϵ \text{T}_{\text{E}}\text{X}` can be used equally well in `dvi mode` (`tex` \rightarrow `dvi` \rightarrow `ps` \rightarrow `pdf`) or in `pdf mode` (`tex` \rightarrow `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:



3 Acknowledgment

This package has seen the light of day thanks to the assistance of Christophe CASSEAU, who had the idea. I thank him for his help before writing the code and for the tests he carried out.

I also want to warmly thank Theo HOPMAN for offering to translate this manual into English.

Operation of chemfig

This part is devoted to describing the most common features of chemfig. The user will find here explanations sufficient to draw most molecules. The presentation of features is done from a theoretical angle, and the goal of this part is not to draw real molecules but to give the user a formal description of the functionality of chemfig. The “Advanced usage”, page 26, will be more practical and will illustrate advanced features for the most demanding uses. It will also highlight methods of building real molecules, page 37. Finally, the last part will give examples of molecules and the code used to draw them.

1 The `\chemfig` macro

The macro `\chemfig` has the following syntax

$$\backslash\text{chemfig}[\text{list of } \langle\text{keys}\rangle=\langle\text{values}\rangle]\{\langle\text{molecule code}\rangle\}$$

The optional argument in square brackets sets the parameters used for this molecule. It should be noted that the parameters are only modified for the current molecule and will be restored to their previous values after the macro has been executed. To permanently modify parameters, the macro `\setchemfig{\langle\text{key}\rangle=\langle\text{values}\rangle}` should be used.

Here is the complete list of parameters as well as their default values. It should be noted that the $\langle\text{keys}\rangle$ from [scheme debug](#) included to the end of the list concern reaction schemes and make no sense in the optional argument of the macro `\chemfig` where they are simply ignored:

$\langle\text{keys}\rangle$	default $\langle\text{values}\rangle$	$\langle\text{keys}\rangle$	default $\langle\text{values}\rangle$
<code>chemfig style</code>	$\langle\text{empty}\rangle$	<code>gchemname</code>	true
<code>atom style</code>	$\langle\text{empty}\rangle$	<code>schemestart code</code>	$\langle\text{empty}\rangle$
<code>bond join</code>	false	<code>schemestop code</code>	$\langle\text{empty}\rangle$
<code>fixed length</code>	false	<code>scheme debug</code>	false
<code>cram rectangle</code>	false	<code>compound style</code>	$\langle\text{empty}\rangle$
<code>cram width</code>	1.5ex	<code>compound sep</code>	5em
<code>cram dash width</code>	1pt	<code>arrow offset</code>	1em
<code>cram dash sep</code>	2pt	<code>arrow angle</code>	0
<code>atom sep</code>	3em	<code>arrow coeff</code>	1
<code>bond offset</code>	2pt	<code>arrow style</code>	$\langle\text{empty}\rangle$
<code>double bond sep</code>	2pt	<code>arrow double sep</code>	2pt
<code>angle increment</code>	45	<code>arrow double coeff</code>	0.6
<code>node style</code>	$\langle\text{empty}\rangle$	<code>arrow double harpoon</code>	true
<code>bond style</code>	$\langle\text{empty}\rangle$	<code>arrow label sep</code>	3pt
<code>baseline</code>	0pt	<code>arrow head</code>	-CF
<code>debug</code>	false	<code>+ sep left</code>	0.5em
<code>cycle radius coeff</code>	0.75	<code>+ sep right</code>	0.5em
<code>stack sep</code>	1.5pt	<code>+ vshift</code>	0pt
<code>show cntcycle</code>	false		
<code>autoreset cntcycle</code>	true		

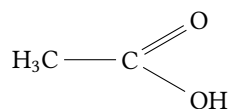
The $\langle\text{molecule code}\rangle$ contains instructions for drawing the molecule according to a syntax that will be explained in this document. There are no restrictions on the characters accepted in the code:

- all catcode 11 or 12 characters, i. e. upper and lower-case letters, numbers, mathematical operators (+ - * / =), punctuation marks whether active or not (. , ; : ! ? ' ' " |), parenthesis and brackets;
- more special characters such as "~", "#"¹ as well as "^" and "_" which have their normal mathematical mode properties;
- spaces, but these are ignored by default because the atoms are composed in mathematical mode;
- the "{" and "}" braces that have their normal behavior as group markers or macro argument delimiters;
- macros.

2 Groups of atoms

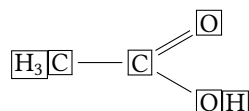
Drawing a molecule consists inherently of connecting groups of atoms with lines. Thus, in the molecule $\text{O}=\text{O}$, there are two groups of atoms, each consisting of a single atom "O".

However, in this molecule



there are four groups of atoms: "H₃C", "C", "O" and "OH". For reasons which we shall see later, chemfig splits each group into single atoms. Each atom extends up to the next capital letter or one of these special characters: \square \square \square \square \square \square \square \square . chemfig ignores all characters inside braces when splitting groups into atoms.

Therefore the first group of atoms "H₃C" is split into two atoms: \square and \square . In terms of chemistry, of course, these are not real atoms; H₃, for example, consists of three hydrogen atoms. In what follows the word atom refers to chemfig's definition. Thus chemfig sees the preceding molecule as follows:



A space is ignored when at the beginning of a group of atoms.

3 First atom's role

It is important to understand that the placement of the entire molecule depends on the first atom placed, i.e. the first atom of the first group of atoms. For this first atom, its tikz anchor "base east" is placed on the baseline of the current line (drawn in gray in the examples of this manual).

Influence of the first atom
<pre>\chemfig{A-B}\qqquad \chemfig{-B}\qqquad \chemfig{A^1-B}</pre> <div style="float: right; margin-top: 10px;"> $A \text{---} B \text{---} B \text{---} A^1 \text{---} B \text{---}$ </div>

To set an arbitrary vertical offset or place a group of atoms on the baseline, use the *baseline* key (see page 30).

4 Different types of bonds

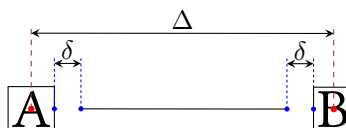
For chemfig, bonds between two atoms are one of nine types, represented by the characters \square , \square , \square , \square , \square , \square , \square , \square and \square :

¹To avoid that # is doubled when the macro \chemfig is in the argument of a macro, instead of #, the macro \# or the macro \CFhash can be used.

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 `<key>` `double bond sep=<dim>` adjusts the spacing between the lines in double or triple bonds. This spacing is 2pt by default.

We must understand that when a bond is made between two atoms, these atoms are contained within invisible rectangular boxes. The centers of these two rectangles are separated by an adjustable distance Δ called the “interatomic distance”. Furthermore, bonds do not connect to the exact edges of the rectangles: a length δ , also adjustable, separates the edges of the rectangles and the beginning and end of the bond line. The rectangular boxes are made visible in the diagram below to help understanding.



The `<key>` `atom sep = <dim>` adjusts the interatomic distance Δ . This setting, like all other settings, affects all the following molecules.

Interatomic distance	
<code>\chemfig[atom sep=2em]{A-B}\par</code>	A — B
<code>\chemfig[atom sep=50pt]{A-B}</code>	A ——— B

The `<key>` `bond offset = <dim>` sets the spacing δ between the bond line and the atom. Its default value is 2pt.

Trimming bonds	
<code>\chemfig[bond offset=0pt]{A-B}\par</code>	A — B
<code>\chemfig[bond offset=5pt]{A-B}</code>	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	
<code>\chemfig{A-B==C}</code>	A — B = C

The `<key>` `bond style = <tikz code>` sets the style for all the bonds drawn thereafter. The `<tikz code>` is empty by default. To custom a single bond, see page 12.

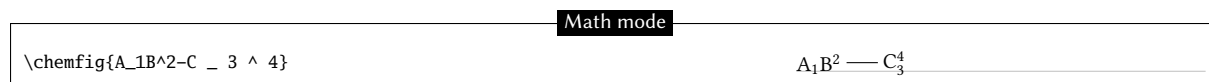
Style of bonds	
<code>\chemfig[bond style={line width=1pt,red}]{A-B=C> D<E>:F}</code>	A — B = C ▷ D ◄ E ... F

The spacing δ for just one bond can be specified with either the character `#`, the macro `\#` or `\CFhash`. It is important to notice that if the macro `\chemfig` is in the argument of a macro, `#` must *not* be used and in that case, `\#` or `\CFhash` must be preferred.

The token `#`, `\#` or `\CFhash` 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	
<code>\setchemfig{bond offset=4pt}</code>	A — B — C
<code>\chemfig{A-B-C}\par</code>	A — B — C
<code>\chemfig{A-#(,0pt)B-C}\par</code>	A — B — C
<code>\chemfig{A-B-#(0pt)C}\par</code>	A — B — C
<code>\chemfig{A-#(,0pt)B-#(0pt)C}</code>	A — B — C

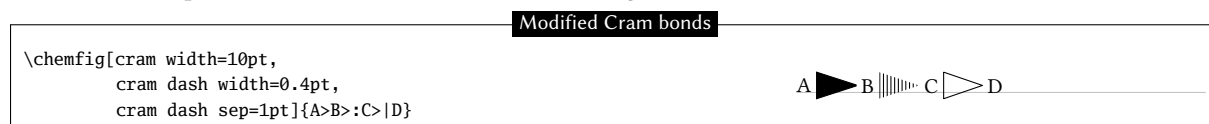
By default, all atoms within groups of atoms are typeset in math mode (spaces are ignored). They may therefore contain math mode specific commands such as subscripts or superscripts²:



There are settings specifically for Cram bonds:

- `cram width = <dim>` is the size of the base of the triangle, and is 1.5pt by default;
- `cram dash width = <dim>` is the thickness of the dots, and is 1pt by default;
- `cram dash sep = <dim>` is the spacing between the dots, and is 2pt by default.

Here is an example where the three dimensions are changed:



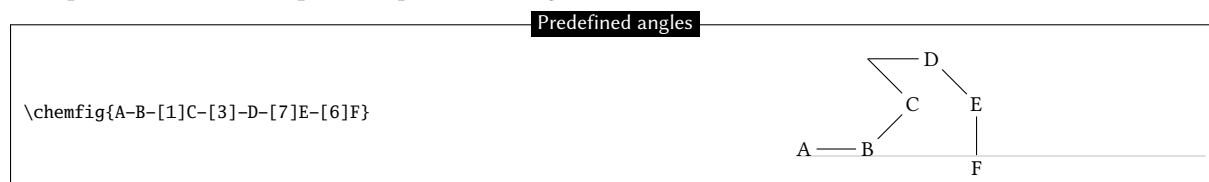
5 Bond angle

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.

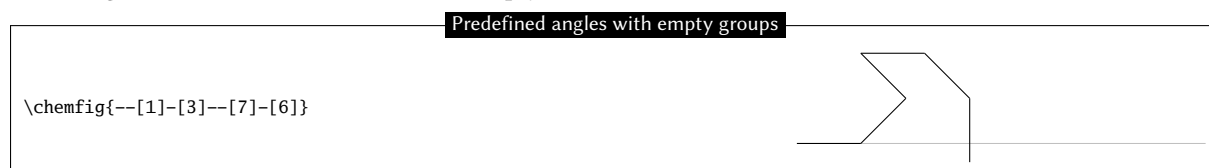
There are several ways of specifying the bond angle.

5.1 Predefined angles

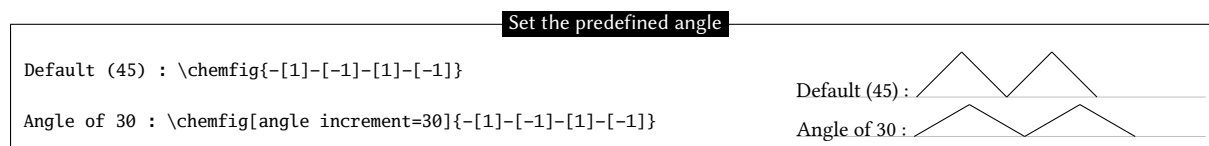
When the angle field contains an integer, this represents the angle the bond makes relative to the horizontal, in multiples of 45°. For example, [0] specifies an angle of 0°, [1] is 45°, and so on.



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



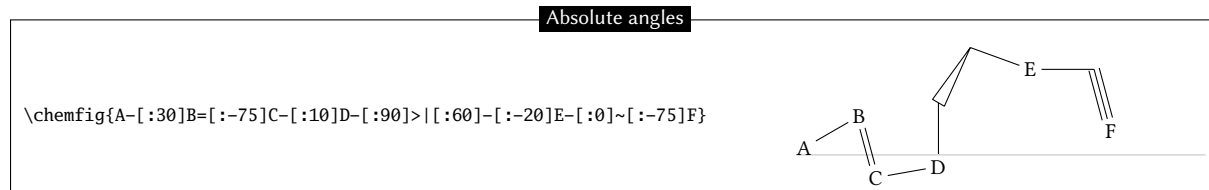
The `<key> angle increment = <angle>` sets the default angle used to calculate the angle of a bond:



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

5.2 Absolute angles

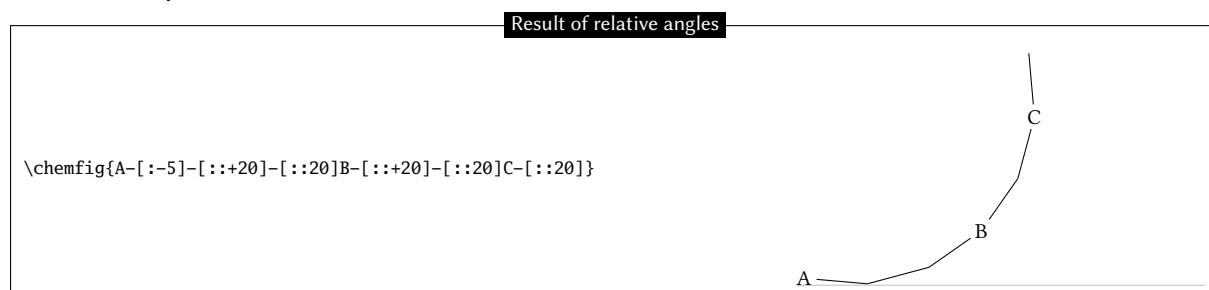
If one wishes to specify an angle in degrees relative to the horizontal, then the optional angle field must take this form: `[:<absolute angle>]`. If necessary, the `<absolute angle>` is reduced to the interval `[0, 360)`:



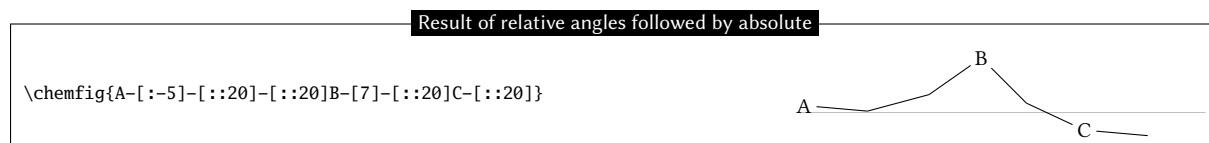
5.3 Relative angles

It is often useful to specify a bond angle relative to the preceding bond. This syntax must 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° :

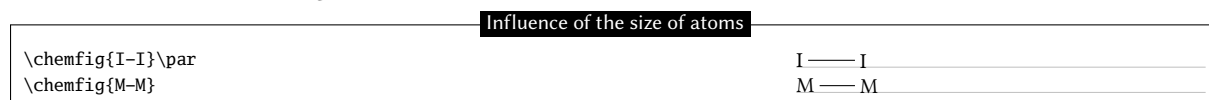


One can “break” a chain of relative angles by putting an absolute or predefined angle where desired. Here, atom “B” is followed by a bond at an absolute angle of 315° .

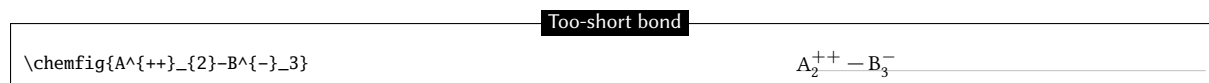


6 Length of a bond

Rather than speaking of length of a bond, we should use the term interatomic spacing. In effect, only the interatomic spacing is adjustable with `atom sep` as we have seen on page 8. 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:

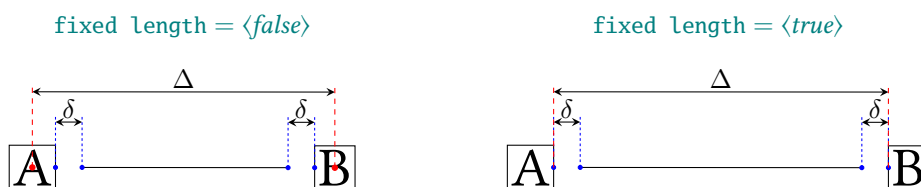


This aspect of the size of atoms becomes particularly acute when the atom involves subscripts or superscripts. In this example, the bond is extremely short, to the point of confusion with a negative sign `-`:

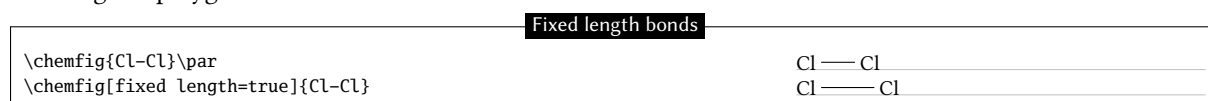


It is important to note that the exponent $-$ is *put inside braces*. If this were not done, `chemfig` would stop the atom on this character, which is a bond character. The atom would then be “B[^]”, which would lead to unexpected results.

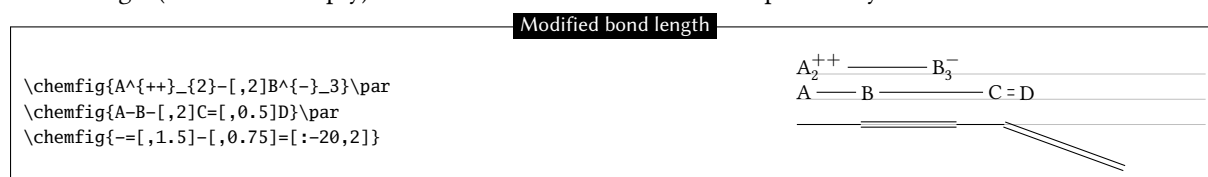
It is possible to change the behavior of `chemfig` about the interatomic spacing. Indeed, when the `\chemfig` macro is immediately followed by a star, the `\key{atom sep}` no longer defines the distance between the centers of atoms, denoted Δ , but *length of the bonds*. Consequently, the bonds have fixed lengths while the distance between the centers of the atoms is variable and depends on their size. Here is the diagram on page 8 and what becomes with the two boolean values of key `fixed length`:



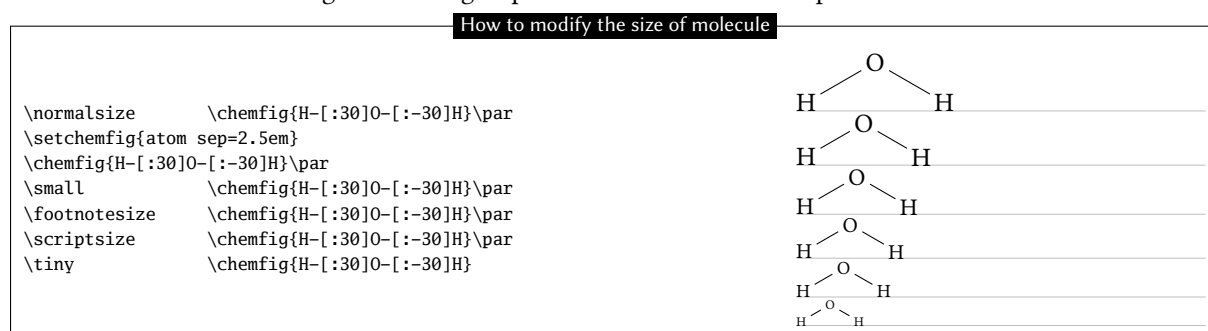
In rings, even when `fixed length = <true>`, the default behavior is restored for the bonds of the cycle, in order to draw regular polygons.



Especially with the default behavior, to avoid too short bonds, it is sometimes necessary to increase (or perhaps reduce) the interatomic distance. 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 `\key{atom sep}`, possibly on both, being careful to confine these changes within a group if we want to limit the scope:



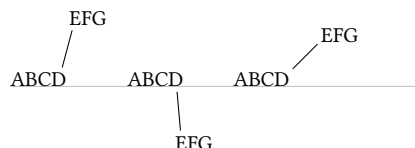
7 Departure and arrival atoms

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.

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

Default atom connections

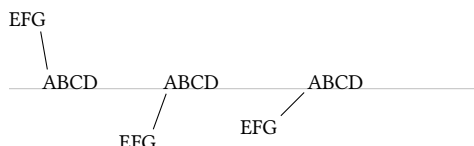
```
\chemfig{ABCD-[ :75]EFG}\quad
\chemfig{ABCD-[-85]EFG}\quad
\chemfig{ABCD-[1]EFG}
```



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

Default atom connections

```
\chemfig{ABCD-[ :100]EFG}\quad
\chemfig{ABCD-[-110]EFG}\quad
\chemfig{ABCD-[5]EFG}
```



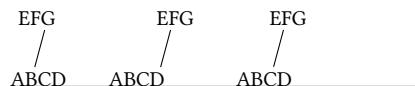
One may sometimes want the bond partners to be atoms other than those determined by `chemfig`. The departure and arrival atoms can be set with the optional bond argument by writing:

$$[, , \langle integer\ 1 \rangle , \langle integer\ 2 \rangle]$$

where $\langle integer\ 1 \rangle$ and $\langle integer\ 2 \rangle$ are the numbers of the desired departure and arrival atoms. These atoms must exist, otherwise an error message will be given.

Specified atom connections

```
\chemfig{ABCD-[ :75,,2,3]EFG}\quad
\chemfig{ABCD-[ :75,,2]EFG}\quad
\chemfig{ABCD-[ :75,,3,2]EFG}
```



8 Customization of bonds

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

$$[, , , , \langle tikz\ code \rangle]$$

This $\langle tikz\ code \rangle$ 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:

Passing tikz code

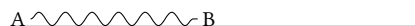
```
\chemfig{A-[,,,red]B}\par
\chemfig{A-[,,,dash pattern=on 2pt off 2pt]B}\par
\chemfig{A-[,,,line width=2pt]B}\par
\chemfig{A-[,,,red,line width=2pt]B}
```



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:

Wavy bonds

```
\chemfig{A-[ ,3,,,decorate,decoration=snake]B}
```



Cram bonds ignore thickness and dash settings.

9 Connecting bonds

By default, the lines representing single bonds do not connect, which can be too visible and unsightly when the line widths are large.

For those who find this “ugly³”, it is now possible connect the single bonds with a slightly increased compilation time. The boolean $\langle key \rangle$ `bond join = $\langle boolean \rangle$` macro enables this feature when $\langle true \rangle$ and disables it when $\langle false \rangle$, which is the better behavior, set by default.

³See <http://tex.stackexchange.com/questions/161796/ugly-bond-joints-in-chemfig> detokenize

Connecting bonds

```
\setchemfig{bond style={line width=3pt}}
\chemfig{-[1]-[7]} et
\chemfig[bond join=true]{-[1]-[7]}
```



The problem is even more obvious with Cram bonds when they connect to a single bond or another Cram bond. Here again, setting `bond join` to `<true>` makes for more aesthetically pleasing connections.

Connecting Cram bonds

```
\chemfig{<[: -20]>[:50]}\qqquad
\chemfig[bond join]{<[: -20]>[:50]}
\medbreak
\setchemfig{cram width=5pt}
\chemfig{<[: -45]-[:30,,,line width=5pt]>[: -10]}\qqquad
\chemfig[bond join]{<[: -45]-[:30,,,line width=5pt]>[: -10]}
```



It is important to note the tow following points:

- `bond join` is `<true>`, the Cram bond is drawn without an outline: it will therefore appear slightly thinner (the importance depending on the `<line width>` parameter);
- `bond join` has no effect between a simple bond and a Cram bond when the 2 vertices of the base of the triangle of the Cram bond are outside the space delimited by the 2 parallel edges of the simple bond. Mathematically, it is the case when $d \cos \alpha > w$, where d is the value of `<cram width>`, w is the width of the single bond and α is the angle between the 2 bonds.

Connecting Cram bonds

```
\setchemfig{cram width=5pt, bond join}
\chemfig{-[,,,line width=3pt]>[:30]}\qqquad
\chemfig{-[,,,line width=3pt]>[:65]}
```



10 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`.

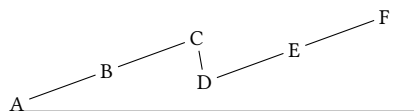
These default values can be changed for the whole molecule by beginning the molecule code with

$$[<angle>,<coeff>,<n1>,<n2>,<code tikz>]$$

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:

Overriding default values

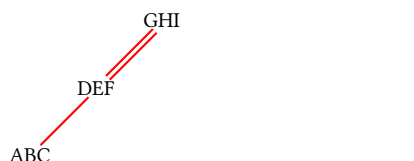
```
\chemfig{[:20,1.5]A-B-C-[: -80,0.7]D-E-F}
```



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:

Default values

```
\chemfig{[1,1.5,2,2,red,thick]ABC-DEF=GHI}
```

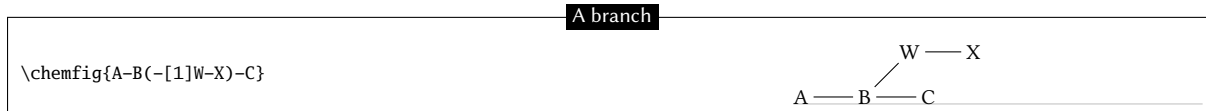


11 Branches

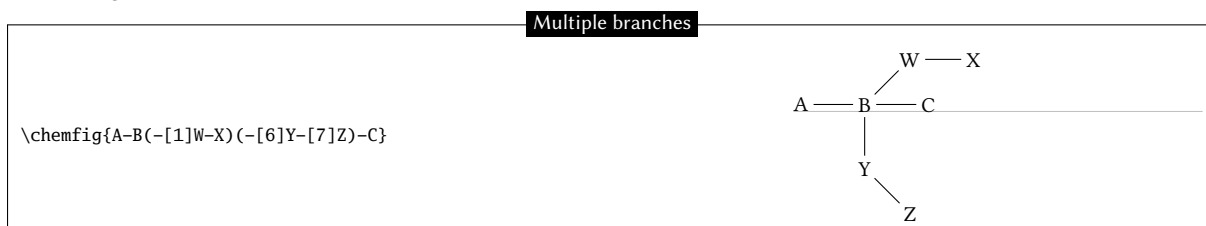
11.1 Principle

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 sub-molecule which will be attached to the atom.

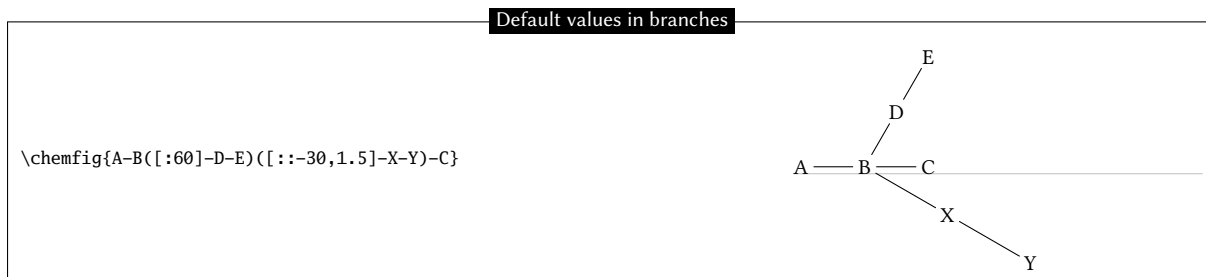
In this example, the sub-molecule “-[1]W-X” will be attached to atom “B”:



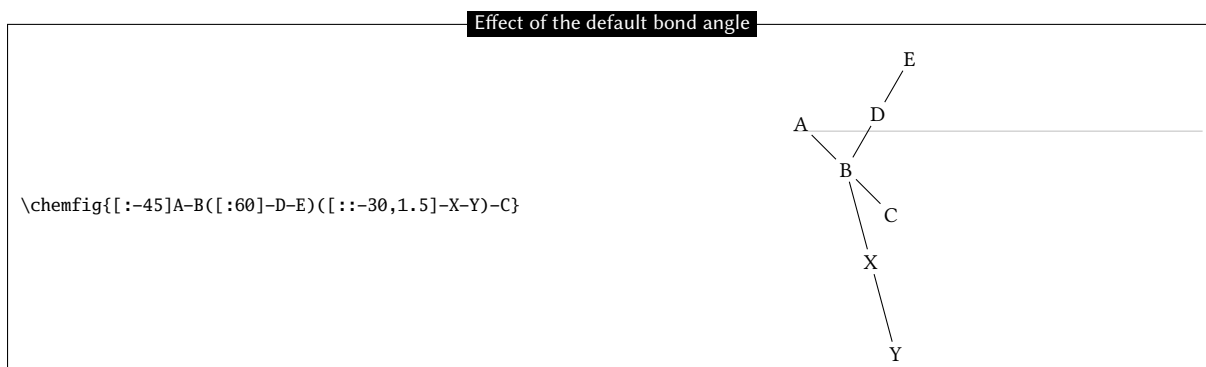
There can be several sub-molecules which are to be attached to the same atom. Just have several parentheses containing the code for each sub-molecule:



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:



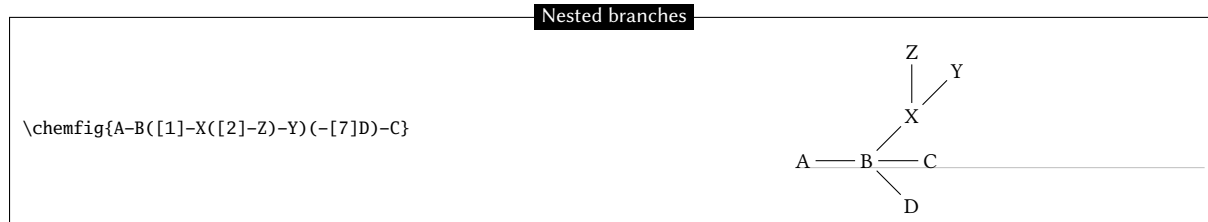
Observe what happens if, at the beginning of the main molecule, one writes “[:-45]”:



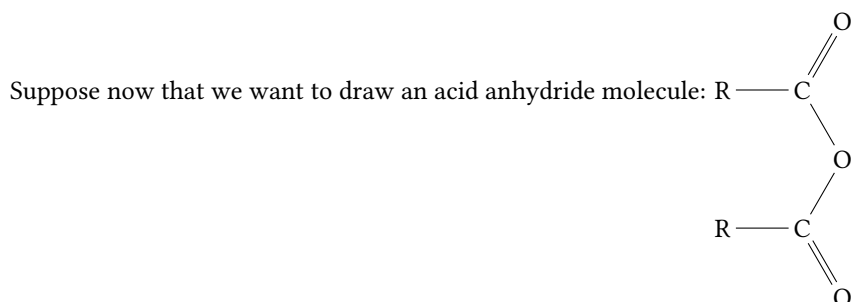
We see that the angle between the bond B-C and the bond B-X stays at 30° because it is a relative angle for the sub-molecule “-X-Y”. By contrast, the branch “-D-E” stays inclined at 60° to the horizontal, and does not follow the rotation given by the -45° angle at the beginning; this is expected because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

11.2 Nesting

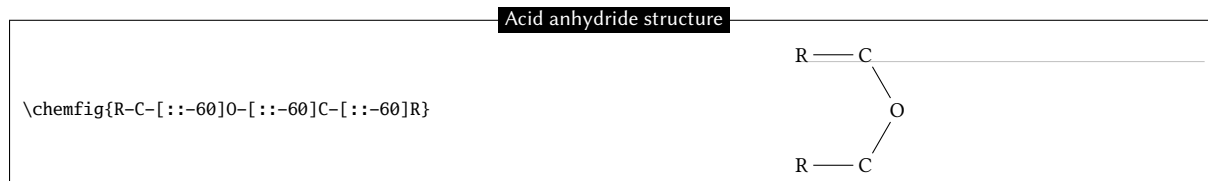
Sub-molecules may be nested, and the rules seen in the preceding paragraphs stay in force:



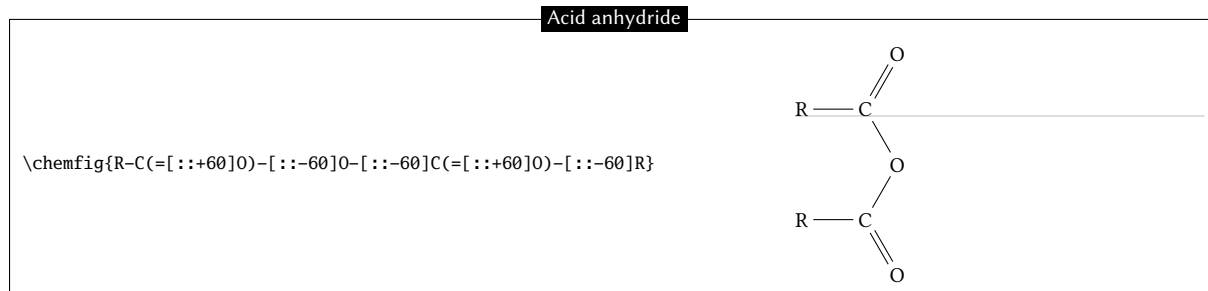
11.3 Method



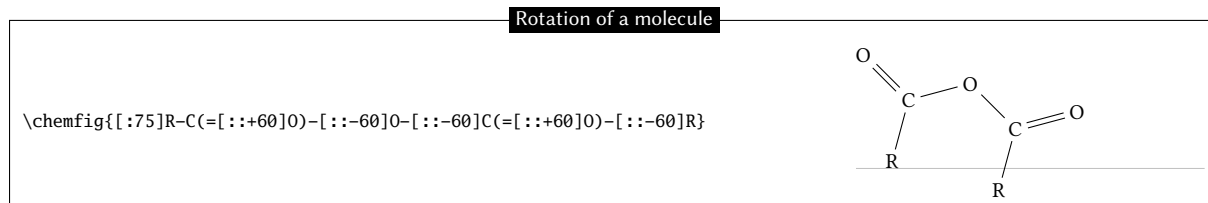
The best way to get this is to find the longest chain. Here, for example, we can draw the chain $R-C-O-C-R$ taking into account angles and using only relative angles:



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



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



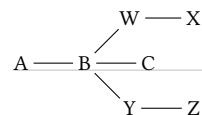
12 Connecting distant atoms

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

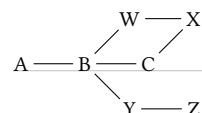
```
\chemfig{A-B(-[1]W-X)(-[7]Y-Z)-C}
```



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

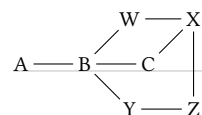
```
\chemfig{A-B(-[1]W-X?)(-[7]Y-Z)-C?}
```



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:

$$?[\langle name \rangle, \langle bond \rangle, \langle tikz \rangle]$$

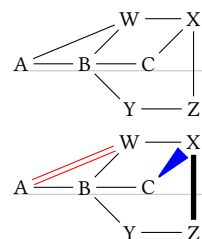
where each field takes its default value if it is empty:

- The $\langle name \rangle$ 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.
- $\langle bond \rangle$ 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 7 for the bond codes.)
Second, the field can be one of the bond character codes, provided that this character is *between braces*.
- $\langle tikz \rangle$ 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:

Multiple distant bonds

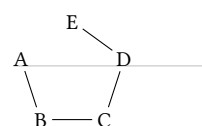
```
\chemfig{A?[a]B(-[1]W?[a]-X?[b])(-[7]Y-Z?[b])C?[b]}\par\medskip
\chemfig{A?[a]B(-[1]W?[a,2,red]-X?[b])(-[7]Y-Z?[b,1,{line width=2pt}])C?[b,{>},blue]}
```



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:

An incomplete ring

```
\chemfig{A-[:72]B-C-[:72]D-[:144]E}
```

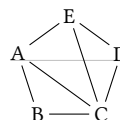


Then we must do this:

⁴This is not exactly right. Actually all the characters that can be put between `\csname... \endcsname` are allowed.

Multiple distant bonds

```
\chemfig{A?[a]-[:72]B-C?[a]?[b]-[:72]D-[:144]E?[a]?[b]}
```



13 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.

13.1 Syntax

`chemfig` can easily draw regular polygons. The idea is to attach a ring to an $\langle atom \rangle$ outside the ring with this syntax:

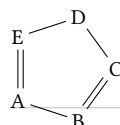
$$\langle atom \rangle * \langle n \rangle (\langle code \rangle)$$

$\langle n \rangle$ is the number of sides of the polygon and the $\langle code \rangle$ 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”:

5-ring

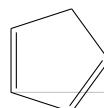
```
\chemfig{A*5(-B=C-D-E=)}
```



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-ring with empty groups

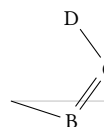
```
\chemfig{*5(-----)}
```



A ring can be incomplete:

Incomplete 5-ring

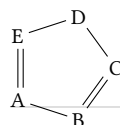
```
\chemfig{*5(-B=C-D)}
```



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:

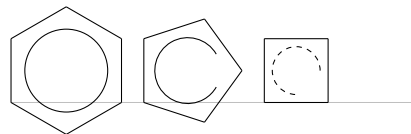
$$\langle atom \rangle ** [\langle angle 1 \rangle, \langle angle 2 \rangle, \langle tikz \rangle] \langle n \rangle (\langle code \rangle)$$

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

- $\langle angle\ 1 \rangle$ and $\langle angle\ 2 \rangle$ 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;
- $\langle tikz \rangle$ 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(-----)}
```



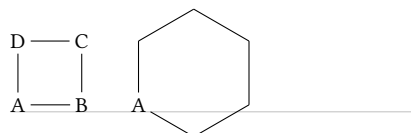
13.2 Angular position

13.2.1 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

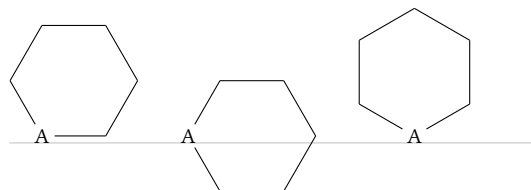
```
\chemfig{A*4(-B-C-D-)}\quad\chemfig{A*6(-----)}
```



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(-----)}
```



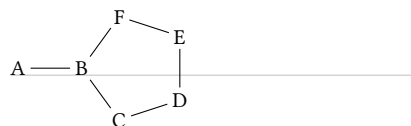
13.2.2 After a bond

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:

Bond ending on a ring

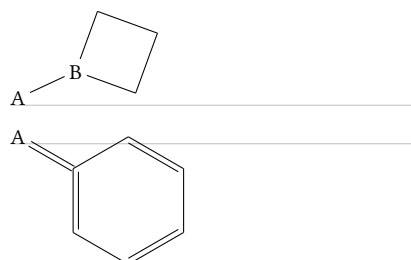
```
\chemfig{A-B*5(-C-D-E-F-)}
```



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

Bonds ending on a ring

```
\chemfig{A-[:25]B*4(----)}\vskip5pt
\chemfig{A[:-30]*6(=----)}
```

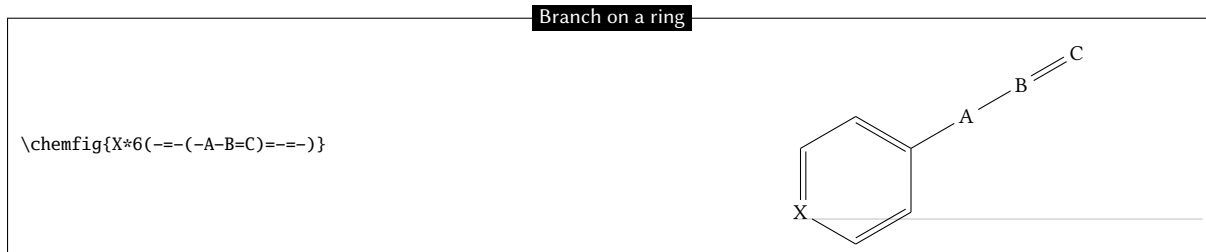


13.3 Branches on a ring

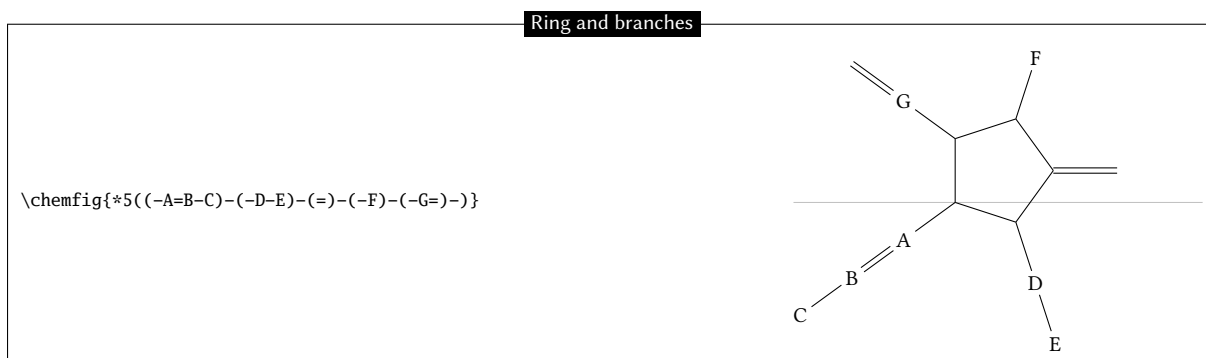
To have branches attached to the vertices of a ring, we use the syntax we have already seen:

$$\langle atom \rangle (\langle code \rangle)$$

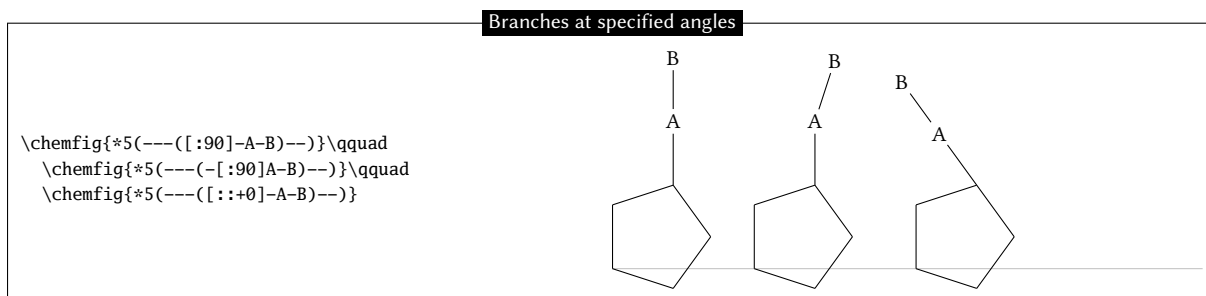
where the $\langle code \rangle$ is that of the sub-molecule and the $\langle atom \rangle$ is at the vertex. Unique to rings, the default angle of the sub-molecule is not 0° but is calculated so that it will bisect the sides leaving the vertex:



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

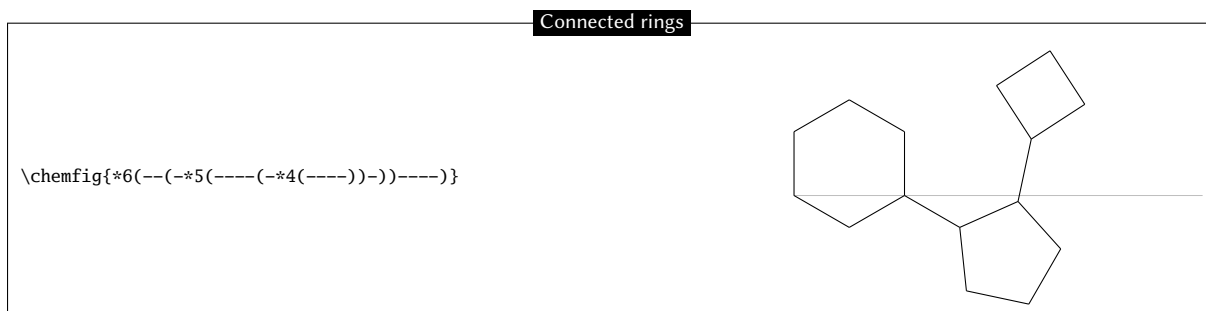


If one wants the bond leaving a vertex not to be the bisector of its sides, one can tinker with the optional global parameter or the optional bond parameter:



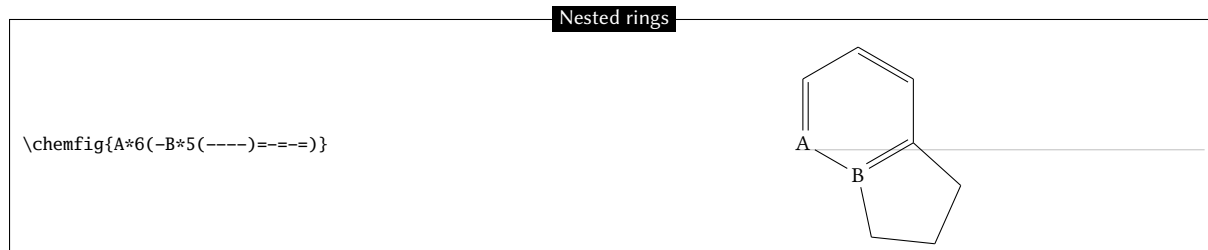
It is worth noting that in the third example, where a relative angle of 0° was given, the bonds of the branch are drawn in line with the preceding bond in the ring. This is the rule on page 10 which specified that the reference angle was that of the bond last drawn.

We can now connect together rings with bonds:



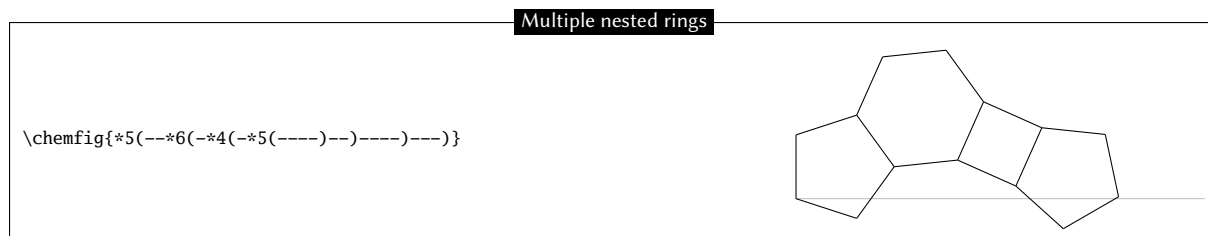
13.4 Nested rings

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:

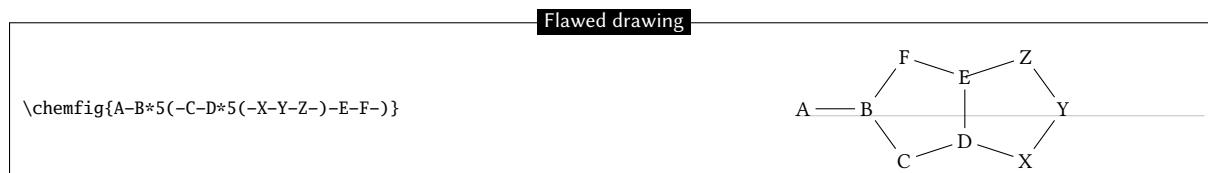


Note that the ring which is going to be attached to the main ring has an angular position such that two of the rings' sides coincide. In addition, the 5-ring has only four bonds “----”. In effect, the fifth will be useless because it is the second side of the 6-ring, which has already been drawn.

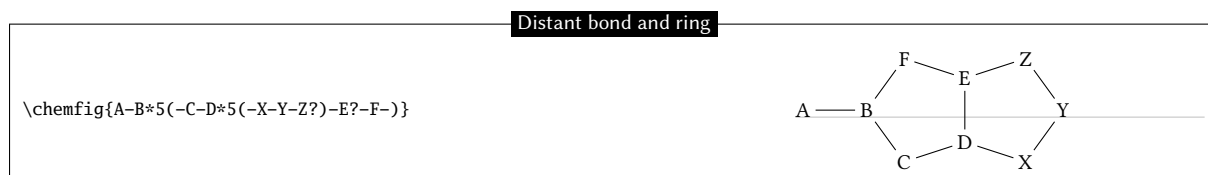
It is quite possible to glue multiple rings together:



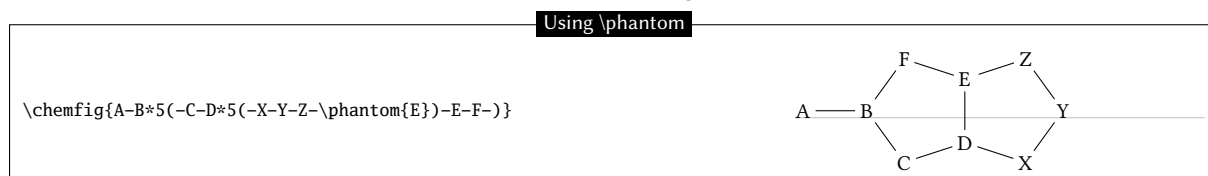
There is a case where a trick must be used. It can be seen in this example that the fourth side of the second 5-ring just passes through the center of atom “E”.



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:



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

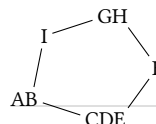


13.5 Rings and groups of atoms

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

Ring and groups of atoms

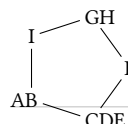
```
\chemfig{AB*5(-CDE-F-GH-I-)}
```



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-)}
```



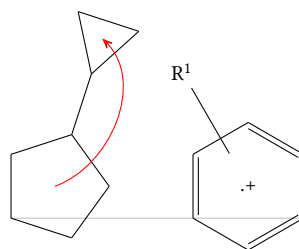
13.6 Center of rings

Each ring has at its center a node of zero dimension whose name is `centrecycle⟨n⟩` where $\langle n \rangle$ is the number of the ring, (the rings are numbered in the order in which they are drawn). It is possible to display the number of each ring by setting the boolean `show cntcycle` to true.

The default true boolean `autoreset cntcycle` resets the ring counter at the beginning of each molecule to 0 (i.e., each time `\chemfig` is executed).

Centre of rings

```
\chemfig{*5(---(-*3(---))--)}
\chemmove{\draw[red](cyclecenter1)to[out=20,in=-45](cyclecenter2);}
\qqquad
\chemfig{*6(-----=)}
\chemmove{%
  \node[at=(cyclecenter1)]{.+.}
  \node[at=(cyclecenter1),shift=(120:1.75cm)](end){\printatom{R^1}};
  \draw[-,shorten <=0.5cm](cyclecenter1)--(end);
}
```



14 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 “ $\langle name \rangle$ ” and the $\langle coeff \rangle$, which must be between 0 and 1, determines where the node is located on the bond. If “`@{⟨name⟩}`” is used, the $\langle coeff \rangle$ 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 “ $\langle name1 \rangle$ ” to the node named “ $\langle name2 \rangle$ ”:

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

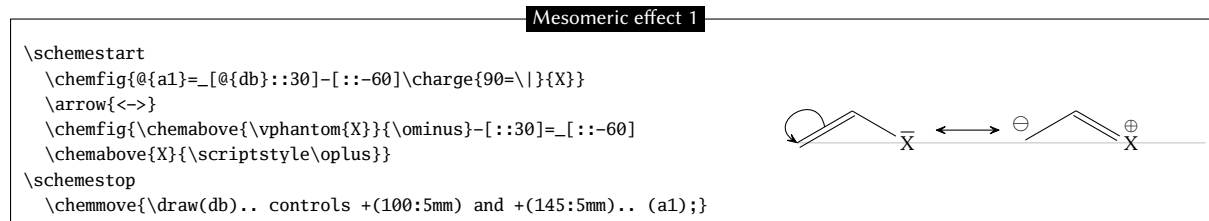
```
\chemmove[⟨opt⟩]{\draw[⟨tikz opt⟩](⟨name1⟩)⟨tikz link⟩(⟨name2⟩);}
```

The optional argument $\langle opt \rangle$ 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 $\langle tikz opt \rangle$ and $\langle tikz link \rangle$ instructions are describe in detail in the documentation of the `tikz` package.

14.1 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.

The macros `\schemestart`, `\schemestop`, `\arrow` and `\+` are explained in the chapter IV, starting on page 49.

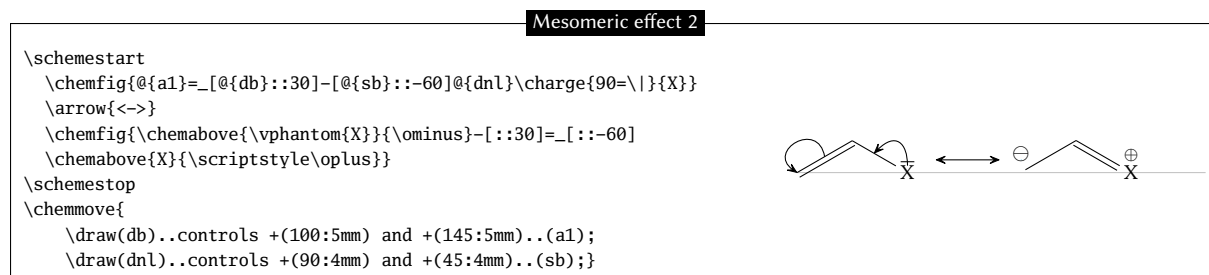


As noted above, there is no comma after the node placed in the optional arguments of a bond; we write `"_=[@{db}::30]"` and not `"_=[@{db}, ::30]"` as one might be tempted to do.

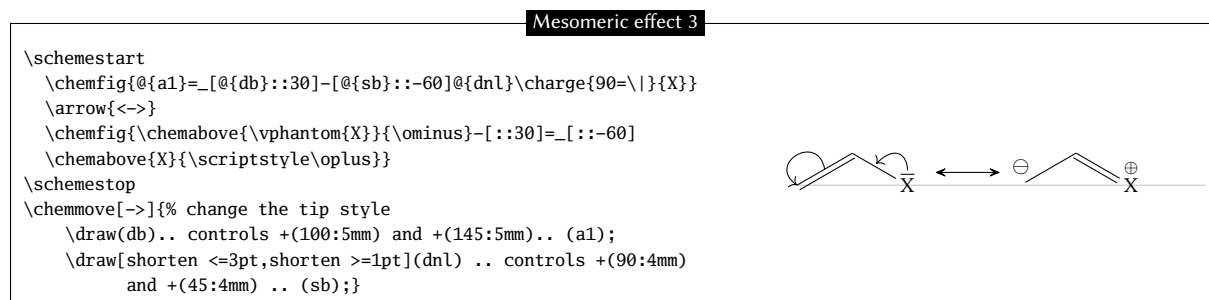
To link the nodes "db" and "a1" we have used the following syntax:

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

For arrows in `\chemmove`, the default tip is "CF". 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").



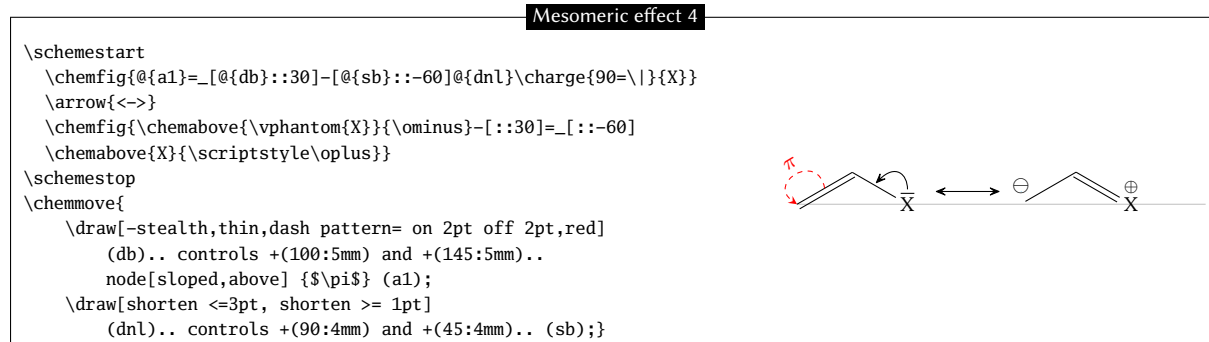
For our new arrow we have set the control points as follows: 4 mm at an angle of 90° from "dnl" and 4 mm at an angle of 45° from "sb". But we are not completely satisfied, since we would like the arrow not to touch the line segment representing the lone pair. To do this we will add some options to our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



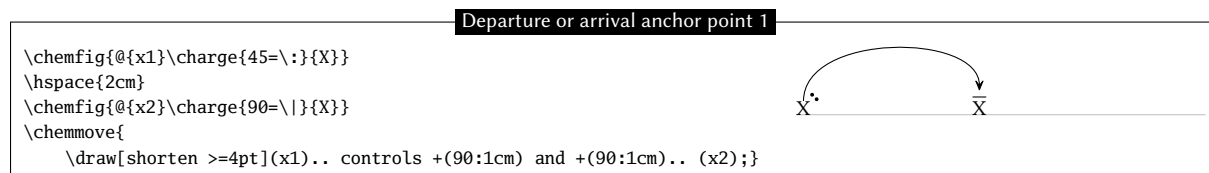
⁶To find all the ways of connecting two nodes with `tikz`, read the documentation for that package.

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.

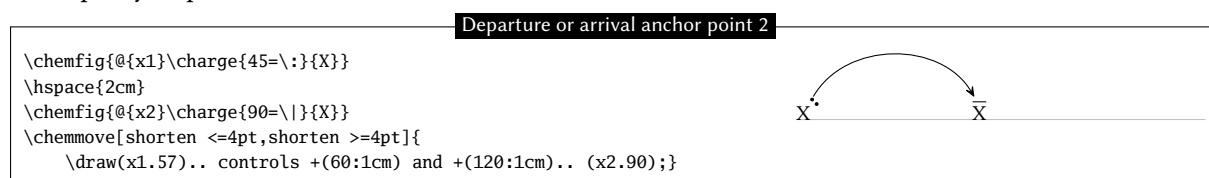
We can use all the power of tikz instructions to modify the style of the arrow. Here we change the head of the arrow leaving the double bound and set it to “-stealth”, and we draw the arrow with a fine dashed red line. We also add the letter π above the middle of the arrow:



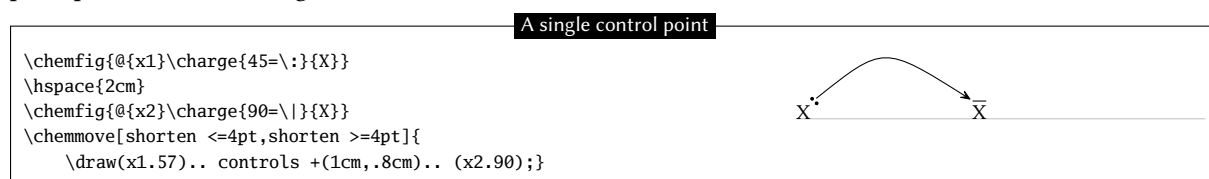
In the following example, we’ll see how to indicate the position of the departure or arrival anchor points of the arrow. If we write



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 center 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”:



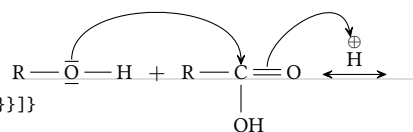
All the graphics drawn by means of the command `\chemmove` are superimposed and will not be included in the bounding boxes. We can see this in the preceding example.

14.2 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.

Esterification: step 1

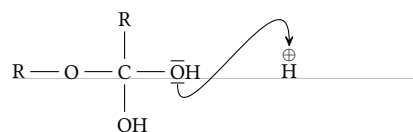
```
\setchemfig{atom sep=7mm}
\schemestart
\chemfig{R-@{dnL}\charge{90=\|,-90=\|}{O}-H}
\+
\chemfig{R-@{atoc}C([6]-OH)=[@{db}]O}
\arrow(.mid east--){<->[\chemfig{@{atoh}\chemabove{H}{\scriptstyle\oplus}}]}
\schemestop
\chemmove[shorten <=2pt]{
\draw(dnL)..controls +(90:1cm)and+(north:1cm)..(atoc);
\draw[shorten >=6pt](db)..controls +(north:5mm)and+(100:1cm)..(atoh);}
```



The use of the `\chemabove{<code>}{<matériel>}` 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.

Esterification: step 2

```
\setchemfig{atom sep=7mm}
\chemfig{R-O-C(-[2]R)(-[6]OH)-@{dnL}\charge{90=\|,-90=\|}{O}H}\hspace{1cm}
\chemfig{@{atoh}\chemabove{H}{\scriptstyle\oplus}}
\chemmove{
\draw[shorten <=2pt, shorten >=7pt]
(dnL)..controls +(south:1cm) and +(north:1.5cm)..(atoh);}
```



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

15 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 centered relative to the molecule, but the `<name>` may not contain multiple paragraphs. As we see in this example: H — O — H, the `<name>` which is displayed under the molecule is

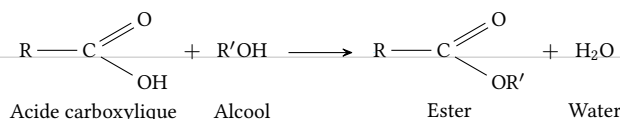
The water molecule: H₂O

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:

Displaying names of molecules

```
\schemestart
\chemname{\chemfig{R-C(-[:30]OH)=[:30]O}}{Acide carboxylique}
\+
\chemname{\chemfig{R'OH}}{Alcool}
\arrow(.mid east--.mid west)
\chemname{\chemfig{R-C(-[:30]OR')=[:30]O}}{Ester}
\+
\chemname{\chemfig{H_2O}}{Water}
\schemestop
\chemnameinit{}
```



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

Name alignment 1

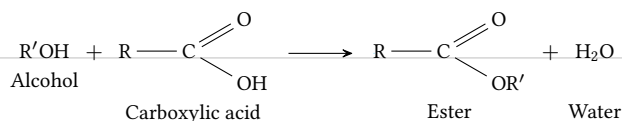
```
\schemestart
\chemname{\chemfig{R'OH}}{Alcohol}
\+
\chemname{\chemfig{R-C(-[:30]OH)=[:30]O}}{Carboxylic acid}
\arrow(.mid east--.mid west)
```



```

\chemname{\chemfig{R-C(-[:30]OR')=[:30]O}}{Ester}
\+
\chemname{\chemfig{H_2O}}{Water}
\schemestop
\chemnameinit{}

```



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

- write `\chemnameinit{\langle deepest molecule \rangle}` 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.

Note that the assignments for the largest depth are global if `gchemname = \langle true \rangle`, which is the default behavior. They are local otherwise.

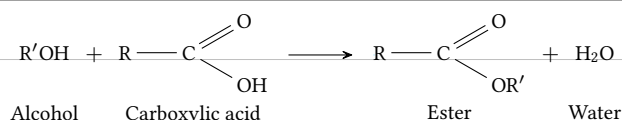
Thus the correct code uses `\chemnameinit` before and after the reaction:

Name alignment 2

```

\chemnameinit{\chemfig{R-C(-[:30]OH)=[:30]O}}
\schemestart
\chemname{\chemfig{R'OH}}{Alcohol}
\+
\chemname{\chemfig{R-C(-[:30]OH)=[:30]O}}{Carboxylic acid}
\arrow(.mid east--.mid west)
\chemname{\chemfig{R-C(-[:30]OR')=[:30]O}}{Ester}
\+
\chemname{\chemfig{H_2O}}{Water}
\schemestop
\chemnameinit{}

```



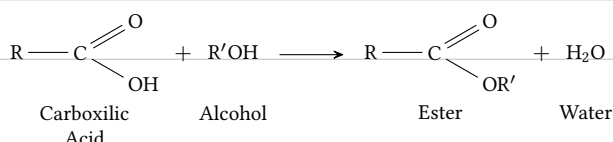
Finally, to write a name on multiple lines, the command `\` encountered in a $\langle name \rangle$ causes a line break⁸:

Name on 2 lines

```

\schemestart
\chemname{\chemfig{R-C(-[:30]OH)=[:30]O}}{Carboxilic\\Acid}
\+
\chemname{\chemfig{R'OH}}{Alcohol}
\arrow(.mid east--.mid west)
\chemname{\chemfig{R-C(-[:30]OR')=[:30]O}}{Ester}
\+
\chemname{\chemfig{H_2O}}{Water}
\schemestop
\chemnameinit{}

```



If `\chemname*{\langle name \rangle}` is written, the macro does not take into account the previous names.

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

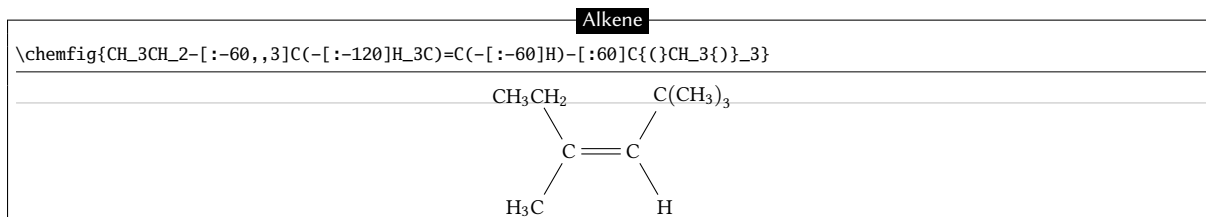
⁸Conversely, the command `\par` is forbidden and causes a compilation error.

Advanced usage

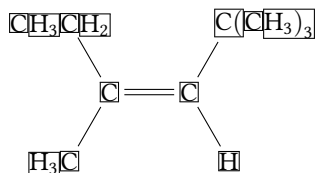
1 Separating atoms

The `separating atom` mechanism described previously extends each atom until the next capital letter or one of the characters `{}|`.

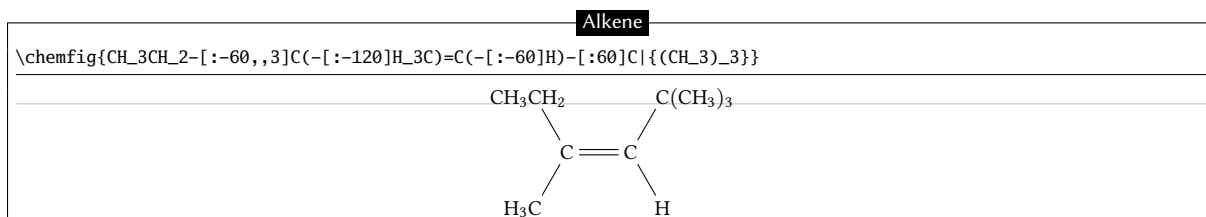
In certain cases this automatic separation produces incorrect atoms which can translate into an imperfect diagram. Consider this example molecule, noting that the “(” character is placed between braces so that `chemfig` doesn't incorrectly create a branch:



We find that the bond which arrives at the carbon atom in the upper right is too short. This happens because, if we apply the `chemfig` rules for separating atoms to the upper right group, the atoms are split in this way: “C{ (”, “C”, “H_3{)}_3”. We now realize that the first atom contains a parenthesis and thus has too great a depth in math mode; we can see this by making the bounding boxes visible:



The character “|” forces splitting of the atom when it is encountered. Thus we can write `C|{(CH_3)_3}` to ensure that `chemfig` separates just two atoms here: “C” and “{(CH_3)_3}”. The problem of the too-short bond is thus solved:



2 Displaying atoms

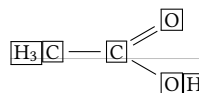
Once a molecule has been split into atoms, the macro `\printatom` is called internally by `chemfig` in order to display each atom. Its sole argument is the code of the atom to be displayed (e.g. “H_3”). By default, this macro enters `math` mode and displays its argument with the math font family “`rm`”. It is defined by the following code:

- `\newcommand*\printatom[1]{\ensuremath{\mathrm{#1}}}` when compiling with \LaTeX
- `\def\printatom#1{\ifmmode\rm#1\else$\rm#1$\fi}` when compiling with $\epsilon\TeX$ ou \ConTeXtX .

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

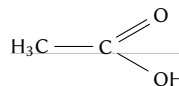
```
\fboxsep=1pt
\renewcommand*\printatom[1]{\fbox{\ensuremath{\mathrm{#1}}}}
\chemfig{H_3C-C(=[:30]O)(-[:-30]OH)}
```



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)}
```



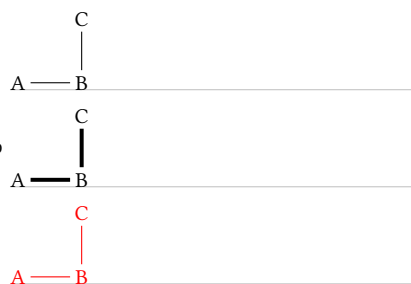
3 Arguments given to tikz

The *key* `chemfig style` contains `tikz` instructions which will be passed to the `tikzpicture` environment in which the molecule is drawn. On the other hand, the *key* `atom style` contains `tikz` instructions which will be executed when each node; these instructions are added to the end of every node/.style<argument>, i.e. after the following instructions: “anchor=base, inner sep=0pt, outer sep=0pt, minimum size=0pt”.

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

Style choice

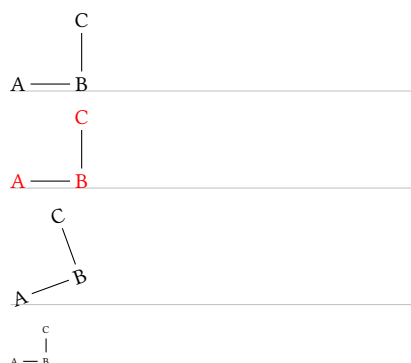
```
\chemfig{A-B-[2]C}\par\medskip
\setchemfig{chemfig style={line width=1.5pt}}\chemfig{A-B-[2]C}\par\medskip
\setchemfig{chemfig style=red}\chemfig{A-B-[2]C}
```



With *node style*, 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
\setchemfig{atom style=red}\chemfig{A-B-[2]C}\par\medskip
\setchemfig{atom style={rotate=20}}\chemfig{A-B-[2]C}\par\medskip
\setchemfig{atom style={scale=0.5}}\chemfig{A-B-[2]C}
```



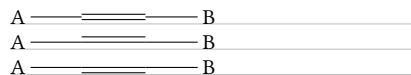
4 Shifted double bonds

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

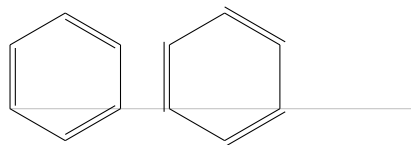
```
\chemfig{A==B}\par
\chemfig{A=^B}\par
\chemfig{A=-B}
```



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

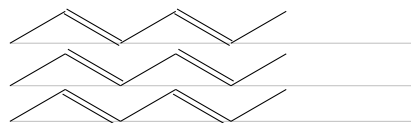
```
\chemfig{*6(====)}\quad
\chemfig{*6(==_==_)}\quad
```



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]}
```

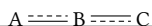


5 Delocalized double bonds

It is sometimes necessary to draw a double bond so that one line would be full and the other dashed. This feature is not hard-coded in `chemfig` since `tikz`, with its “`decorations.markings`” library makes it possible.

Delocalized bonds

```
\catcode'\_ =11
\tikzset{
  ddbond/.style args={#1}{
    draw=none,
    decoration={%
      markings,
      mark=at position 0 with {
        \coordinate (CF@startdeloc) at (0,\dimexpr#1\CF_doublesep/2)
        coordinate (CF@startaxis) at (0,\dimexpr-#1\CF_doublesep/2);
      },
      mark=at position 1 with {
        \coordinate (CF@enddeloc) at (0,\dimexpr#1\CF_doublesep/2)
        coordinate (CF@endaxis) at (0,\dimexpr-#1\CF_doublesep/2);
        \draw[dash pattern=on 2pt off 1.5pt] (CF@startdeloc)--(CF@enddeloc);
        \draw (CF@startaxis)--(CF@endaxis);
      }
    },
    postaction={decorate}
  }
}
\catcode'\_ =8
\chemfig{A-[,,,ddbond={+}]B-[,,,ddbond={-}]C}
```



6 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>}
```

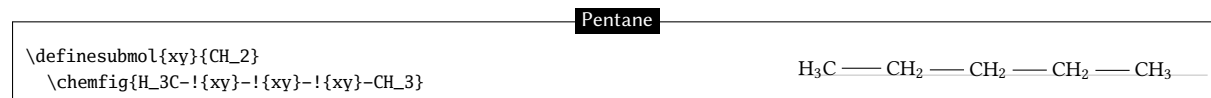
which saves the $\langle code \rangle$ for recall in the code of the molecule via the shortcut “! $\langle name \rangle$ ”. This $\langle name \rangle$ can be:

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

In all cases, if the alias is already defined you should not overwrite it with a new definition using $\backslash definesubmol$. 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:

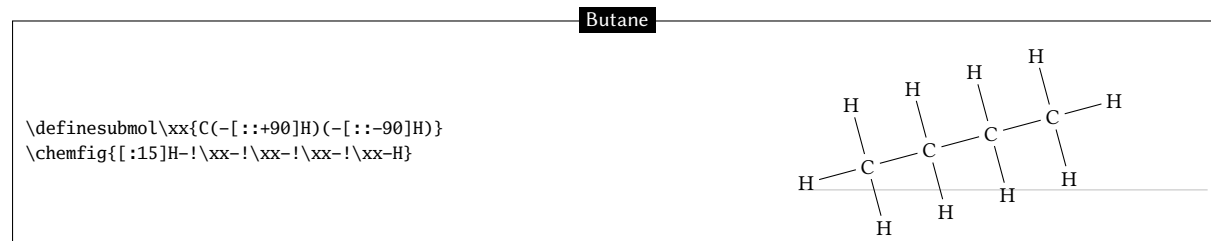
$$\backslash redefinesubmol\{\langle name \rangle\}\{\langle code \rangle\}$$

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



In this case the technique is not very interesting because “! $\langle xy \rangle$ ” 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:

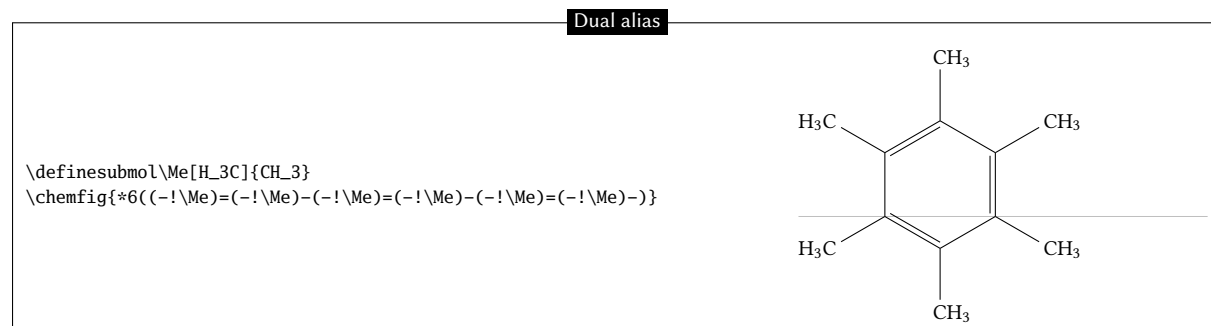


The $\backslash definesubmol$ command takes an optional argument; its syntax is as follows:

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

When the optional argument is present, the alias “! $\langle name \rangle$ ” will be replaced by $\langle code1 \rangle$ 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 $\langle code2 \rangle$.

We will define a control sequence $\backslash Me$ pour “methyl” so that the alias “! $\backslash Me$ ” 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:



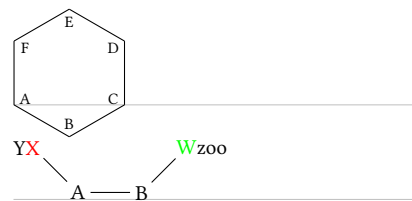
The sub-molecule saved with a $\langle name \rangle$ does not admit an argument when it is called after “!””. To define a sub-molecule admitting one or more arguments, place this $\langle number \rangle$ of arguments just after the $\langle name \rangle$. And the full syntax of $\backslash definesubmol$ is:

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

In the $\langle codes \rangle$, the arguments must appear in their usual form “# $\langle n \rangle$ ” where $\langle n \rangle$ is the argument number.

`\definesubmol with arguments`

```
\definesubmol\X1[-,-0.2,,draw=none]{\scriptstyle#1}  
\chemfig{*6((!\X A)-(!\X B)-(!\X C)-(!\X D)-(!\X E)-(!\X F)-)}  
  
\definesubmol{foo}3[#3|\textcolor{#1}{#2}]{\textcolor{#1}{#2}|#3}  
\chemfig{A(-[:135]!\foo{\red}XY)-B(-[:45]!\foo{\green}{W}{zoo})}
```

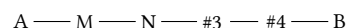


It should be noted that if the $\langle number \rangle$ of arguments is incorrect (negative or greater than 9), an error message will be issued and `chemfig` will consider that the sub molecule does not admit an argument.

Except in cases where the character ”#” is followed by a number between 1 and $\langle number \rangle$ in which case it represents an argument, ”#” are allowed in the sub-molecule codes.

`Use of #`

```
\definesubmol\X2{#1-#2-#3-#(3pt,3pt)#4}  
\chemfig{A-!\X{M}{N}-B}
```



In this example, only #1 and #2 are understood as the arguments of the sub molecule `\X`. The other ”#” are displayed as they are in the molecule (case of #3 and #4) or understood as the character specifying the fine adjustment of the offset of the bonds.

7 Placement of Atoms

7.1 Groups of atoms

In a group of atoms, the atoms are placed one after the other in a well-established order:

- the first one which is placed (which we will call ”reference atom”) is the one on which the bond arrives; in the case of the beginning of the molecule, the atom on the left is the reference atom;
- the atoms to the right of the reference atom are then placed from left to right;
- atoms to the left of the reference atom are finally placed from right to left.

In the group of atoms thus formed, the baselines of each atom are on *the same horizontal line*, in other words, the atoms are all aligned on the same horizontal line.

In the example below whose code would be ”`\chemfig{A[:-60,,3]BCDEF}`” the reference atom of the 2nd group of atoms is ”D” because the bond is requested to arrive on the 3rd atom. Below each atom of this group is the sequence number in which the atom is displayed:



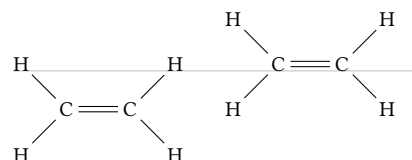
7.2 Vertical alignment

The `baseline` is used to finely control the vertical placement of the molecule in relation to the baseline of the current paragraph.

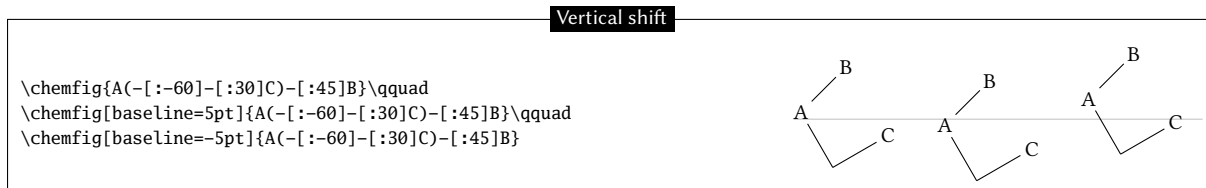
It is set to $\langle Opt \rangle$ by default, and in this case, the first atom encountered (whether empty or not) is the one placed on the baseline of the current paragraph, shown in gray on the examples in this manual. The choice of this first atom therefore conditions the placement of all the others, and often influences the placement of the entire molecule.

`First atom`

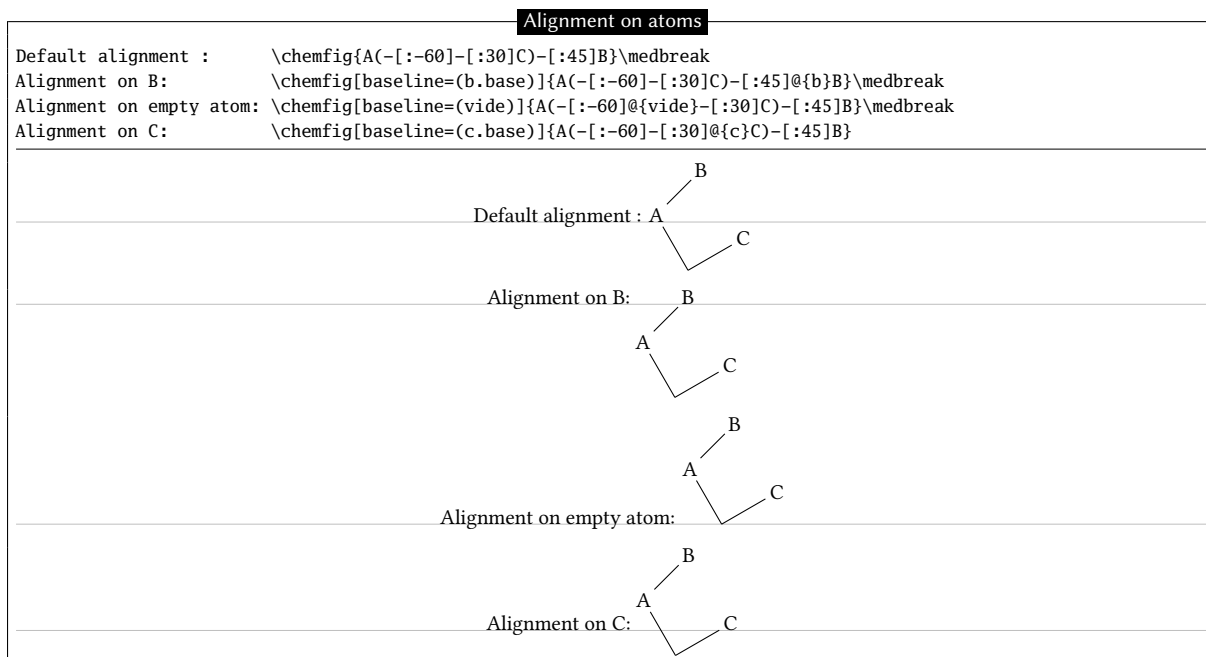
```
\chemfig{H-[7]C(-[5]H)=C(-[1]H)-[7]H}\quad  
\chemfig{C(-[3]H)(-[5]H)=C(-[1]H)-[7]H}
```



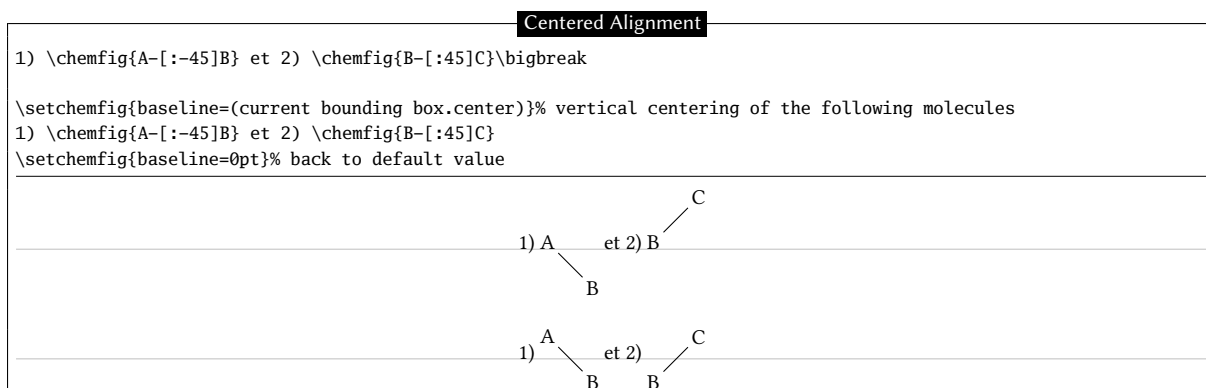
An arbitrary dimension can be specified to vertically shift the molecule by this value with the syntax `baseline = <dimension>`:



With the syntax `baseline = <(name)>` (the name must be in parentheses), we specify that the baseline of the molecule is at the node named `<name>`. The name of the atom can be the one assigned automatically by `chemfig` (of the form `n<a>-`) or a name given by the user using the syntax `@{<name>}` (see page 21).

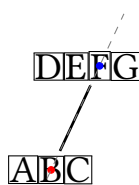


It is possible to name `tikz` nodes. Thus, if we want to vertically center several molecules on the current baseline, we set `baseline` to `<<(current bounding box.center)>>`.



7.3 Bonds between atoms

A bond starting from an atom would, if extended, pass through the centre of its bounding box. The atom placed at the end of the bond has its center of its bounding box in the extension of the bond. Therefore, a bond between two atoms extends through the centers of their bounding boxes, as shown in this example:



This mechanism can create misalignments between groups of atoms that are particularly visible when the bonds are horizontal. Everything works well when the atoms have the same vertical dimensions; however, if a departure atom is high (with exponent) or deep (with subscript) and the arrival atom has a different vertical dimension, the alignment is broken.

Horizontal alignment

```
\Huge\setchemfig{atom sep=2em}
\chemfig{A^1-B-C-D}\quad
\chemfig{E_1-F-G-H}
```



It is surprising that the second atom is correctly aligned when the last two are vertically shifted. This is because `chemfig` adds in front of each arrival atom the `\vphantom` of the departure atom, but without including it in the content of this arrival atom: this `\vphantom` is thus not intended to be reflected on the following atoms. This phenomenon can be shown by making visible the bounding boxes of the atoms where it appears that the atoms "B" and "F" have bounding boxes whose height takes into account the heights of the preceding atom:

Horizontal placement and bounding boxes

```
\Huge\setchemfig{atom sep=2em}
\fbboxsep=0pt
\renewcommand\printatom[1]{\fbbox{\ensuremath{\mathrm{#1}}}}
\chemfig{A^1-B-C-D}\quad
\chemfig{E_1-F-G-H}
```



Since no automatic solution is satisfactory, we can manually get around this problem by creating an departure atom being a "strut" equal to `\vphantom{X}`: thus, the starting atom has a "normal" height and no shift will be reflected on the next group of atoms. A sub-molecule is used here for brevity.

Bypassing vertical placement

```
\Huge\setchemfig{atom sep=2em}
\definesubmol\I{\vphantom{X}}
\chemfig{A^1!\I-B-C-D}\quad
\chemfig{E_1!\I-F-G-H}
```



The disadvantage is that the first bond is too long because the departure atom now has a zero horizontal dimension.

If we are *really sure* to understand the consequences that this will have on the molecule and especially if it lends itself to it, we can redefine `\printatom` so that it forces the atom's bounding box to have fixed vertical dimensions; for example that of "X₁".

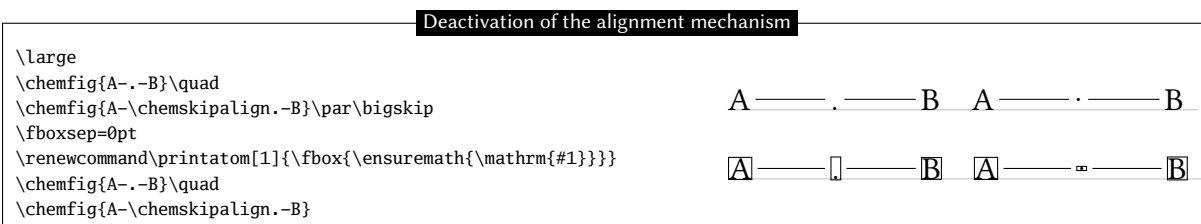
Redefinition of \printatom

```
\Huge\setchemfig{atom sep=2em}
\let\oldprintatom\printatom
\renewcommand\printatom[1]{%
  \begingroup
  \setbox0\hbox{\oldprintatom{X^1_1}}%
  \edef\tmp{\ht0=\the\ht0\relax\dp0=\the\dp0\box0}%
  \setbox0\hbox{\oldprintatom{#1}}%
  \ifnum1\ifdim\ht0=0pt0\fi\ifdim\dp0=0pt0\fi\ifdim\wd0=0pt0\fi<1000
  \tmp
  \fi
  \endgroup
}
```

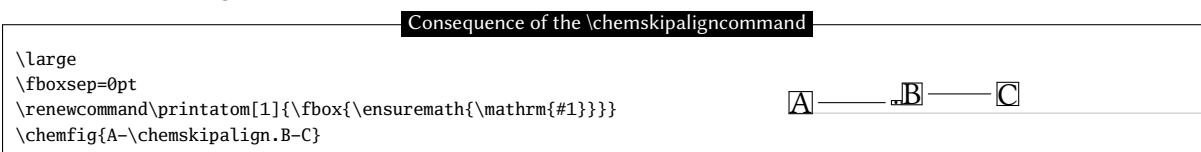



7.4 La macro `\chemskipalign`

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.



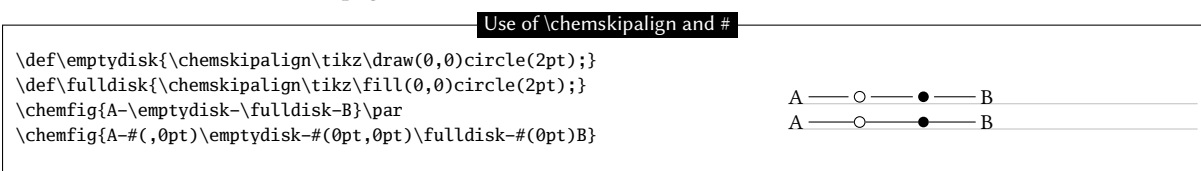
This command is to be used with caution lest the alignment of atoms in the next group be disrupted. In general, all will be well if the group of atoms featuring `\chemskipalign` contains a *single atom* whose height and depth are *less* than those of the preceding and following atoms, and if the preceding 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`”:



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 8.



8 The macro `\charge`

8.1 Overview

The macro `\load`, which requires two mandatory arguments, allows to arrange elements (called `<charges>`) around an `<atome>`; its syntax is

$$\text{\charge}[\langle \textit{general parameters} \rangle][\langle \textit{position} \rangle][\langle \textit{tikz code} \rangle]=\langle \textit{charge} \rangle\{\langle \textit{atom} \rangle\}$$

where:

- the $\langle atom \rangle$ is usually one or two letters, but it can also be empty;
- the $\langle charge \rangle$ is an arbitrary content that will be placed around the atom. Few constraints exist on this content: it can be text (in math mode if needed), or even tikz code or a molecule drawn with chemfig;
- the $\langle general parameters \rangle$ (optional) are a list of key/values specifying the options that this execution of the macro must satisfy. These keys/values are described below;
- the $\langle position \rangle$ is " $\langle angle \rangle:\langle shift \rangle$ ", but it is possible to specify only the $\langle angle \rangle$, in which case, the $\langle shift \rangle$ will be equal to \emptyset pt;
- the optionnal $\langle tikz code \rangle$ sets the options given to the tikz macro $\backslash node$, which places the $\langle charge \rangle$.

8.2 Parameters

The $\langle keys \rangle = \langle values \rangle$ available in the $\langle general parameters \rangle$ are:

$\langle keys \rangle$	default $\langle values \rangle$	Description
<code>debug</code>	false	boolean which, when $\langle true \rangle$, draws the outlines of the nodes receiving the $\langle atoms \rangle$ (in green), the $\langle loads \rangle$ (in blue) and the $\langle charge \rangle$ (in red).
<code>macro atom</code>	<code>\printatom</code>	macro receiving the $\langle atom \rangle$ as argument.
<code>circle</code>	false	boolean which, when $\langle true \rangle$, puts the $\langle atom \rangle$ in a circular node; otherwise, the node is rectangular.
<code>macro charge</code>	$\langle vide \rangle$	macro (e.g., <code>\printatom</code> or <code>\ensuremath</code>) receiving each charge as an argument.
<code>extra sep</code>	1.5pt	node size increment of the $\langle atom \rangle$ to put the $\langle charges \rangle$: it is the value passed to the <code>inner sep</code> of tikz.
<code>overlay</code>	true	boolean which, when $\langle true \rangle$, draws the $\langle charges \rangle$ "overlay", i.e. outside the final bounding box.
<code>shortcuts</code>	true	boolean which, when $\langle true \rangle$, activates the shortcuts " <code>\.</code> ", " <code>\:</code> ", " <code>\ </code> " and " <code>\</code> " to draw Lewis formulas.
<code>lewisautorot</code>	true	boolean which, when $\langle true \rangle$, automatically rotates " <code>\:</code> ", " <code>\ </code> " and " <code>\</code> ".
<code>.radius</code>	0.15ex	radius of the point used to plot " <code>\.</code> " and " <code>\:</code> ".
<code>:sep</code>	0.3em	separation between the two dots of " <code>\:</code> ".
<code>.style</code>	fill=black	tikz style used to draw the " <code>\.</code> " and " <code>\:</code> " dots.
<code>"length</code>	1.5ex	length of the rectangle " <code>\</code> " and the line " <code>\ </code> ".
<code>"width</code>	.3ex	width of the rectangle " <code>\</code> ".
<code>"style</code>	black, line width=0.4pt	tikz style used to draw the rectangle " <code>\</code> ".
<code> style</code>	black, line width=0.4pt	TIKZ style used to draw the line " <code>\ </code> ".

It is possible to set some (or all) of these parameters by running the macro

$$\backslash setcharge\{\langle keys \rangle = \langle values \rangle\}$$

and reset all parameters to their default values with

$$\backslash resetcharge$$

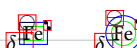
The $\backslash charge$ macro places the $\langle charges \rangle$ out of the bounding box (unless otherwise specified in the $\langle parameters \rangle$) while $\backslash Charge$ places them into the bounding box.

The $\langle angle \rangle$ is the location on the boundary of the node where the $\langle charge \rangle$ is placed. This $\langle angle \rangle$ can be expressed in degrees or it can be a boundary anchor in the sense of tikz, like "south east." The $\langle shift \rangle$ is a TeX-dimension and represents an additional length between the boundary of the node containing the $\langle atom \rangle$ and the place where the $\langle charge \rangle$ is placed. Unless otherwise specified in the $\langle tikz code \rangle$, the *center* anchor of $\langle charges \rangle$.

In the two following examples, `debug` will be set to $\langle true \rangle$ in order to better perceive the changes induced by the modification of the parameters. In addition, the macro $\backslash Charge$ will be used so that the bounding boxes take into account the charges. Here we see the influence of the node shape on the placement of the charges:

Generic example

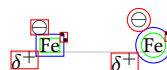
```
\setcharge{debug}
Default then circle:
\Charge{30=\:,120=\ominus$,210=\delta^+}{Fe}\quad
\Charge{[circle]30=\:,120=\ominus$,210=\delta^+}{Fe}
```

Default then circle: 

To place the loads \ominus and δ^+ further away, we can play on the $\langle shift \rangle$ or better, on the anchor: the $\langle angle \rangle$ where the load is placed is stored in the macro `\chargeangle`, so it is wise to choose the anchor `180+\chargeangle`. It is also possible to specify a circular node to place the charge.

Fine positioning

```
\setcharge{debug}
\Charge{30=\:,120:3pt=\ominus$,210:5pt=\delta^+}{Fe}\quad
\Charge{[circle]30=\:,
120[circle,anchor=180+\chargeangle]=\ominus$,
210[anchor=180+\chargeangle]=\delta^+}{Fe}
```



It is important to note that circular nodes have dimensions *sometimes very different* from the "classic" rectangular nodes, especially in terms of horizontal and vertical extent. It is therefore advisable to set $\langle true \rangle$ the boolean key `circle` knowingly.

Circular nodes

```
\chemfig{\charge{90=\.}{N}H_3} : rectangle nodes\smallbreak
\chemfig{\charge{[circle]90=\.}{N}H_3} : circle node
```

$\dot{\text{N}}\text{H}_3$: rectangle nodes _____
 $\dot{\text{N}}\text{H}_3$: circle node _____

8.3 Lewis formula

When `shortcut` is $\langle true \rangle$, the shortcuts `"\.`, `"\:`, `"\|` and `"\"` are active to draw Lewis formulas `"\.`, `"\:`, `"\|` et `"\"`. You can deactivate them at any time with the `\disables shortcuts` macro and reactivate them with `\enables shortcuts`.

When the boolean `shortcut` is $\langle false \rangle$ or the shortcuts have been disabled with `\disables shortcuts`, shortcuts `"\:`, `"\|` and `"\"` are no longer programmed to draw Lewis formulas, so the macros `\chargedot`, `\chargeddot`, `\chargeLine` and `\chargerect` must be used instead.

The key `lewisautorot`, which is $\langle true \rangle$ by default, acts on `"\:`, `"\"` and `"\"` and rotates them.

Autorot

```
\Charge{60=\:,150="}{A} et
\Charge{[lewisautorot=false]60=\:,150="}{A}
```



The customization of Lewis' formulas is done via the macro `\setcharge` or via the optional argument of `\charge` by acting on the keys `.radius`, `:sep`, `.style`, `|style`, `"length`, `"width` and `"style`. It is also possible to modify these keys for each formula with their optional argument which receives a list of `keys = \langle values \rangle`.

Customization

```
\Charge{[.radius=1.5pt,.style={draw=gray}]
45 =\.[{.style={draw=none,fill=red}}],
135 =\.[{.style={draw=none,fill=blue}}],
-45 =\.[{.style={draw=none,fill=green}}],
-135 =\.[{A}\quad
\Charge{
45 =\"["style={draw=red,fill=gray}],
135 =\"["width=3pt,"style={line width=.8pt,draw=blue,fill=cyan}]]{A}
```

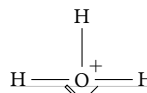


8.4 Integration in chemfig

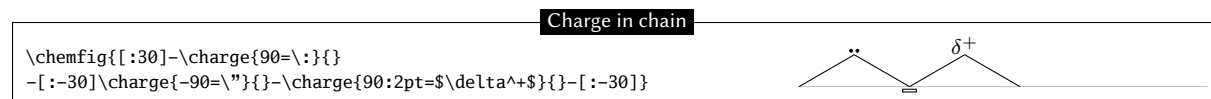
A macro `\charge` can take the place of an atom.

Charge in chemfig

```
\chemfig{H-\charge{45:1.5pt=\scriptstyle+,-45=\|,-135="}{O}(-[2]H)-H}
```



However, `chemfig` has been modified so that the bonds are *joined* when the dimensions of an atom is zero, that is, if its width, height and depth are all `0pt`. This was previously only the case if the atom was empty. This new feature makes it easy to place charges in carbon chains.



9 Stacking

The macros

$$\backslash\text{chemabove}[\langle dim \rangle]{\langle code \rangle}{\langle stuff \rangle}$$

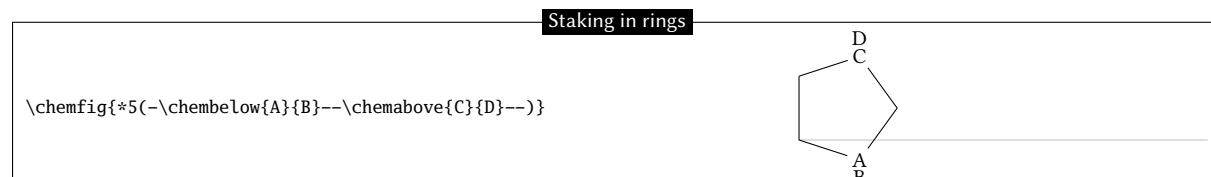
and

$$\backslash\text{chembelow}[\langle dim \rangle]{\langle code \rangle}{\langle stuff \rangle}$$

place the $\langle stuff \rangle$ above and below the $\langle code \rangle$ respectively at a vertical distance $\langle dim \rangle$, without changing the bounding box of $\langle code \rangle$. The optional argument allows, if written, to specify this dimension at each call. If the optional argument is not used, a default size will be taken: its value is $\langle 1.5pt \rangle$ but it can be modified with the $\langle key \rangle$ `stack sep = $\langle dim \rangle$` .

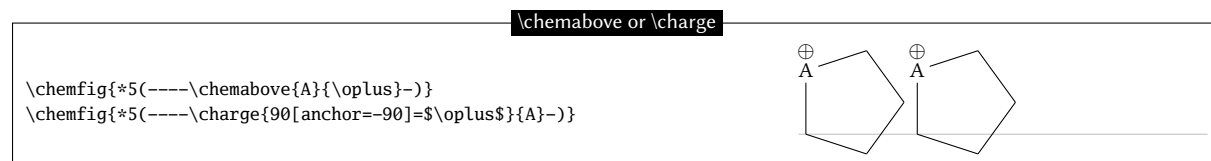
These commands are independent of the macro `\chemfig` and can be used either inside or outside its argument.

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:



The `\Chemabove` and `\Chembelow` commands work in the same way, except that the bounding box takes into account the $\langle stuff \rangle$ placed above or below.

What's the difference between `\chemabove` and `\charge` for placing one item above or below another?



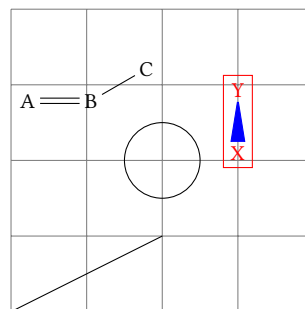
By default, the two macros give very similar results. However, there are differences in their use:

- `\chemabove` and `\chembelow` can only be used in the argument of `chemfig`, which is not the case for `\charge`;
- the `\charge` macro requires `tikz`, whereas `\chemabove` and `\chembelow` use low-level \TeX primitives and are therefore fast and independent of any package.

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

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

```
\begin{tikzpicture}[help lines/.style={thin,draw=black!50}]
  \draw[help lines] (0,0) grid (4,4);
  \draw(0,0) -- (2,1);
  \draw(2,2) circle (0.5);
  \node at (1,3) {\chemfig{A=B-[:30]C}};
  \node[draw,red,anchor=base] at(3,2){\chemfig{X>[2,,,blue]Y}};
\end{tikzpicture}
```

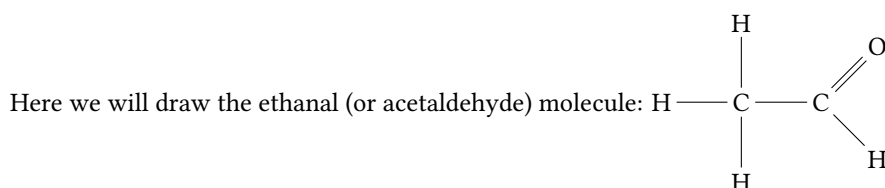


11 Annotated examples

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.

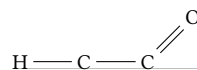
11.1 Ethanal



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 gives a “backbone” of the molecule to which the branches merely have to be added:

Backbone of ethanal

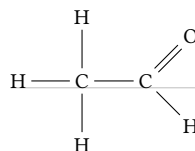
```
\chemfig{H-C-C=[1]O}
```



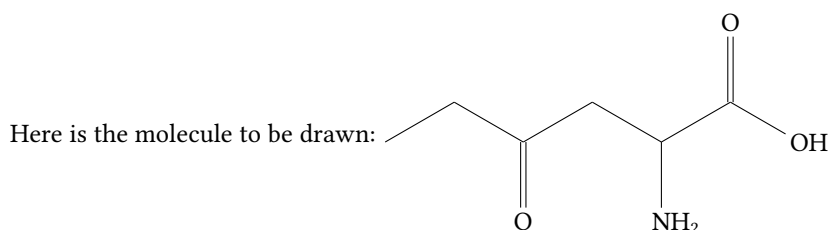
The three hydrogen atoms still have to be placed at the correct orientation with the help of predefined angles. The first is at 90° with the branch “(-[2]H)”, the second at 270° with “(-[6]H)”, and the one on the right at 315° with “(-[7]H)”:

Ethanal

```
\chemfig{H-C(-[2]H)(-[6]H)-C(-[7]H)=[1]O}
```



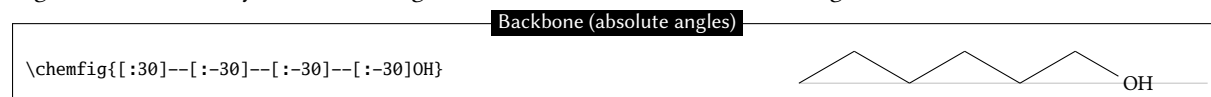
11.2 2-amino-4-oxohexanoic acid



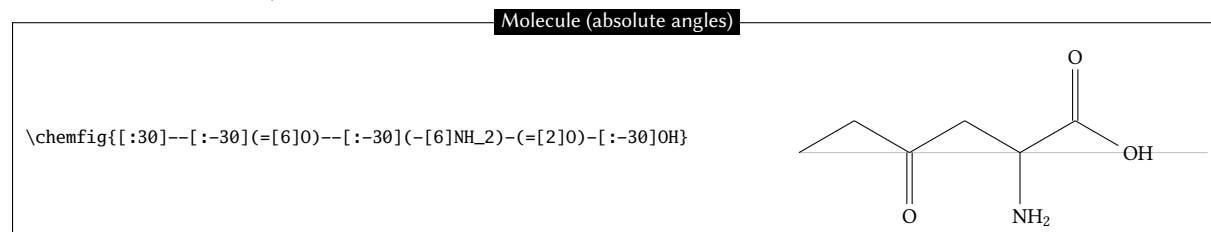
As is often the case for most molecules, there are several methods and for each several different ways of getting the result. Here we will look at four different methods.

11.2.1 Absolute angles

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

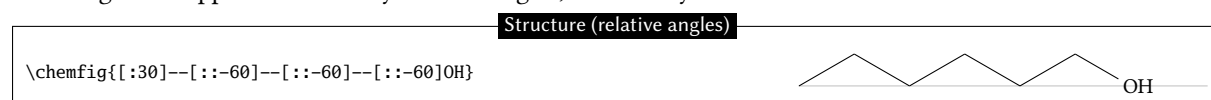


The branches “(=[6]O)”, “(-[6]NH₂)” and “(=[2]O)” still have to be added to the correct vertices:

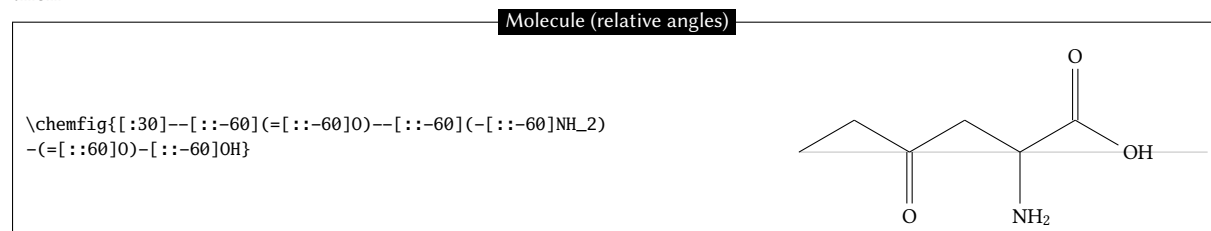


11.2.2 Relative angles

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

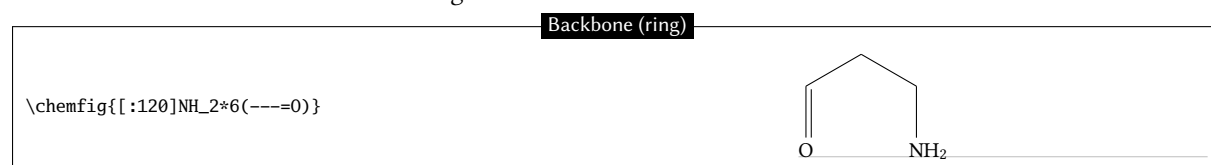


then

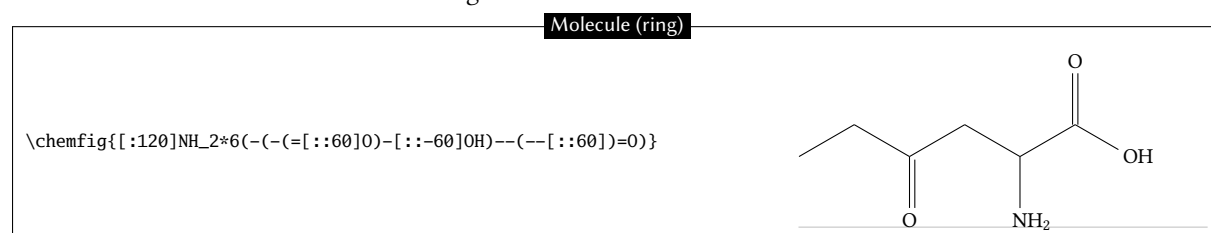


11.2.3 Ring

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:

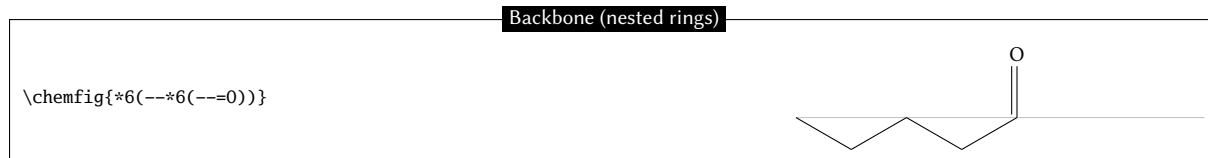


Now the branches must be added to the right vertices:

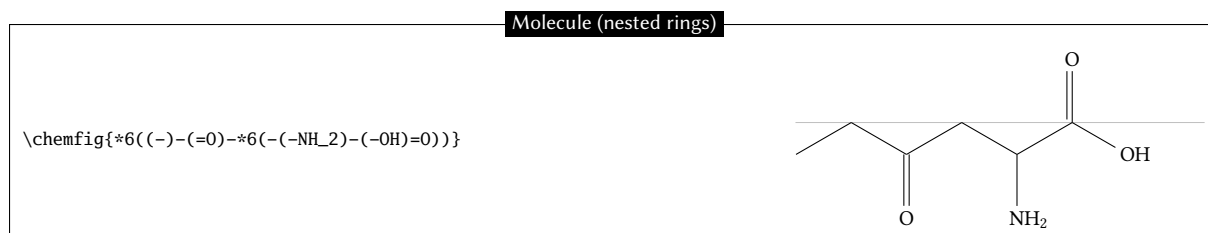


11.2.4 Nested rings

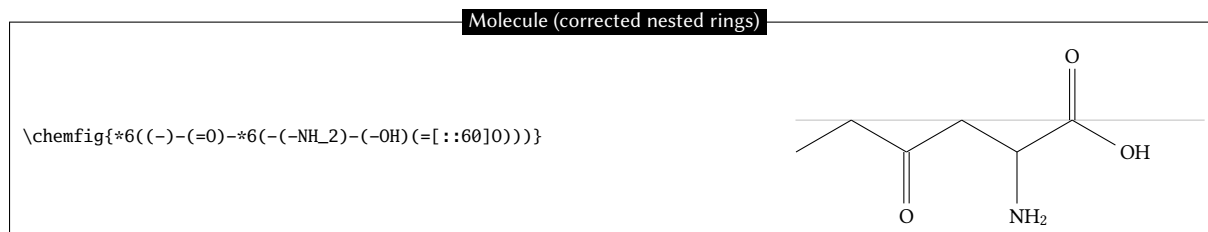
Delving deeper into the ring method, we can also consider nesting incomplete 6-rings. We could start with this backbone:



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-ring⁹ Despite its brevity, this code does not give a perfect drawing. This can of course be corrected by adding a little to the code:

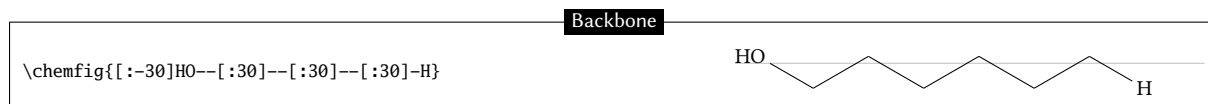


11.3 Glucose

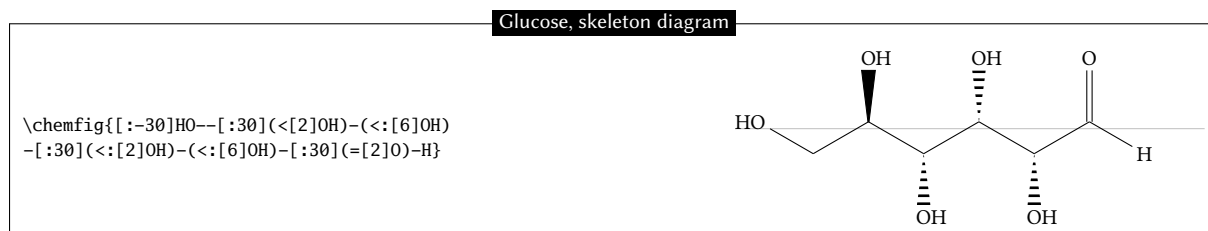
The goal here is to represent the glucose molecule according to several different conventions.

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° :



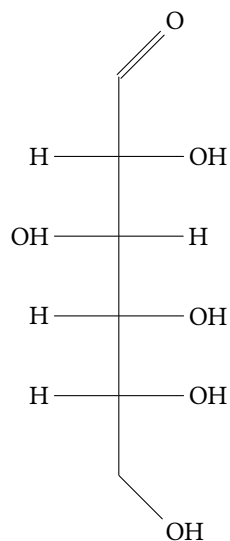
Adding the branches is no problem. We use predefined absolute angles:



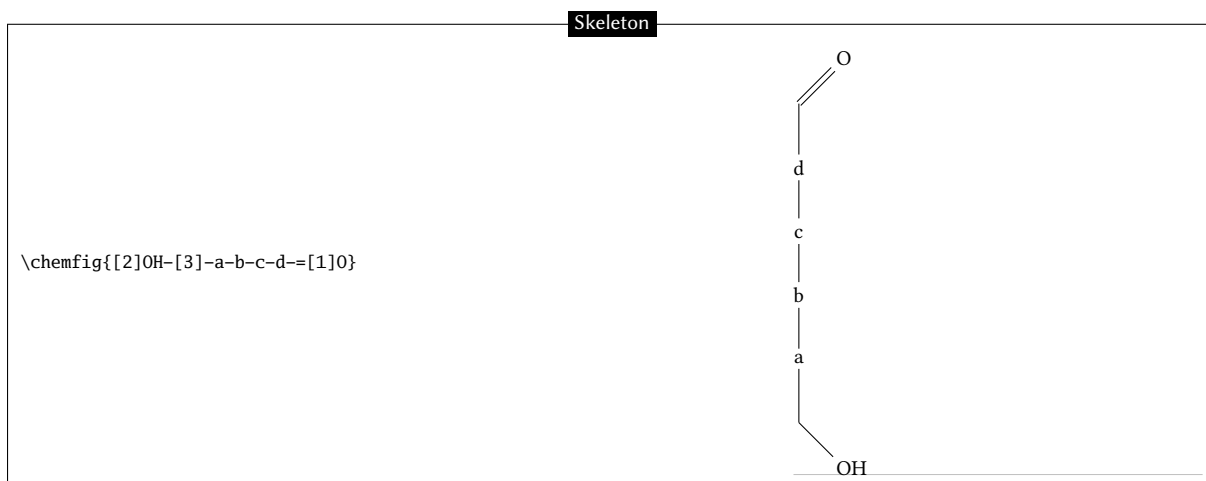
⁹This was also true for the preceding method with one ring.

11.3.2 Fisher projection

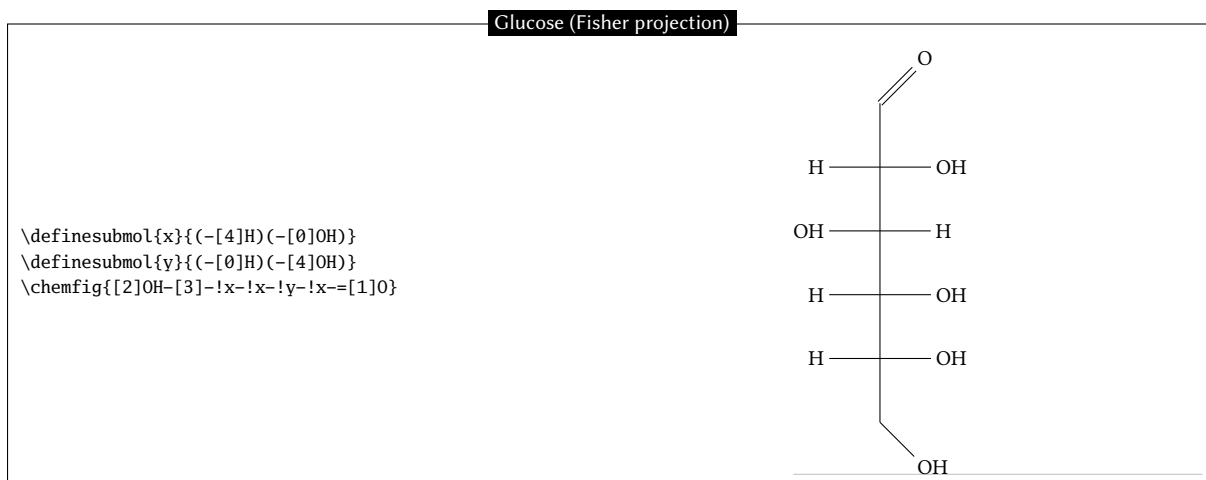
The goal is to get the molecule below:



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:

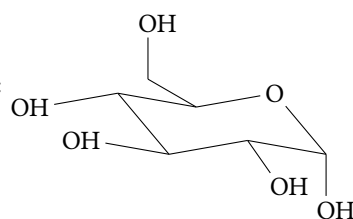


Next we define two aliases for the horizontal bonds and the atoms at their ends. Lets choose “x” which we will put in place of the lower case a, c and d, and “y” which will replace the letter c. Since these aliases are just one character, we do not need braces and can write “!x” instead of “!{x}”:



11.3.3 “Chair” representation

We will depict the α -D-glucose molecule:



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° .

Structure

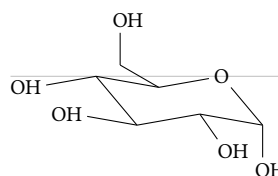
```
\chemfig{?-[:-50]-[:10]-[:-10]-[:130]O-[:190]?}
```



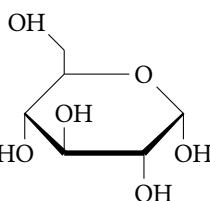
Now we simply add the branches inside parentheses. The angles are chosen to give the best impression of perspective, and some bonds are shortened by a factor of 0.7:

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



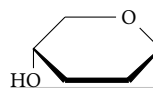
The goal is to depict this D-glucopyranose molecule:

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

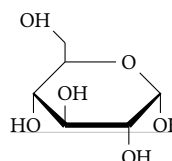
```
\chemfig[cram width=2pt]{HO-[2,0.5,2]?<[7,0.7]-[,,,
line width=2pt]>[1,0.7]-[3,0.7]O-[4]?}
```



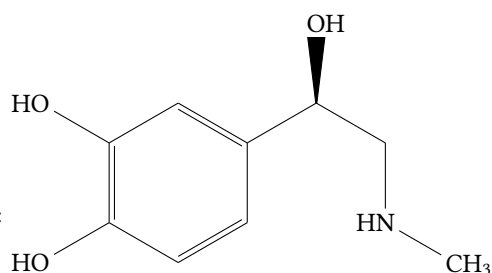
All that needs to be done now is to add the branches at the correct places, giving the right absolute angles and sometimes reducing the length to better give the illusion of perspective:

Projection de Haworth

```
\chemfig[cram width=2pt]{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

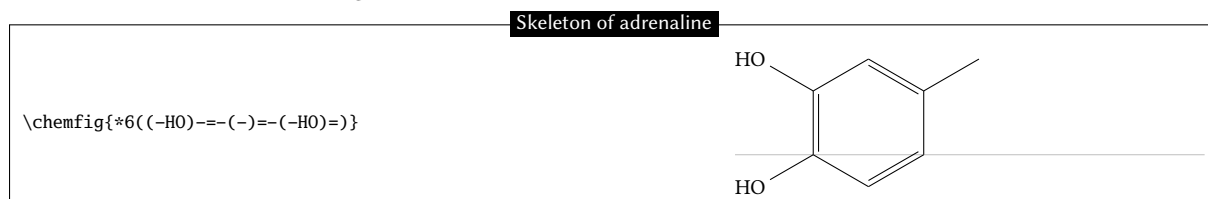


We want to draw the adrenaline molecule:

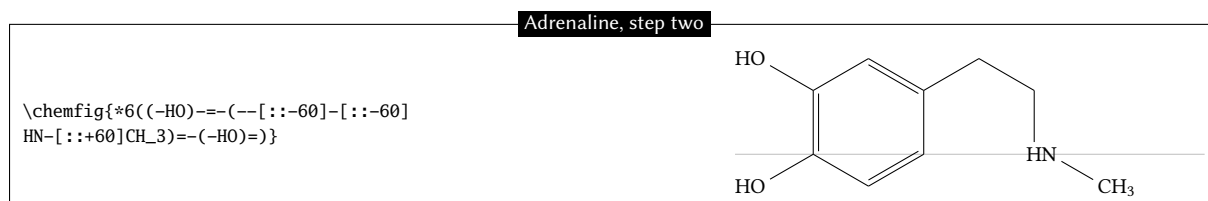
We are going to use two different methods.

11.4.1 Using one ring

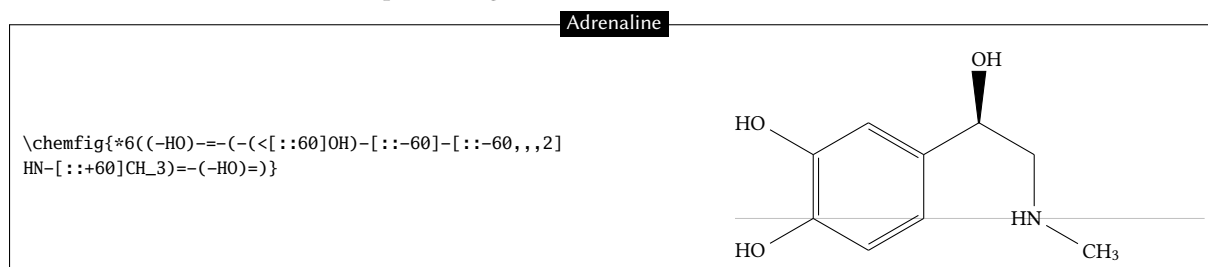
First of all, we start with a 6-ring and we draw the start of the branches which leave it:



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



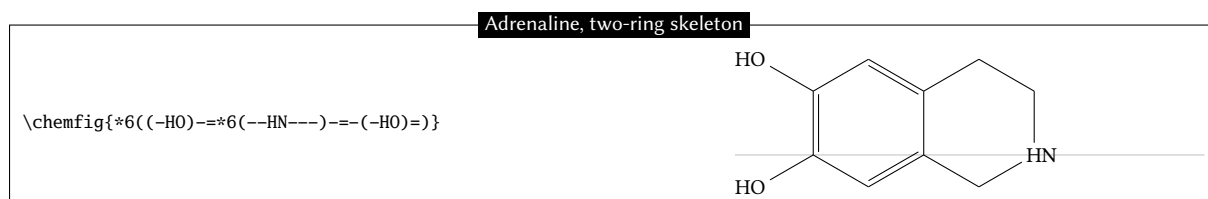
Then we need to add a Cram-bonded OH and indicate that the bond which arrives at "HN" does so on the second atom, i.e., "N". We use the fourth optional argument of the bond:



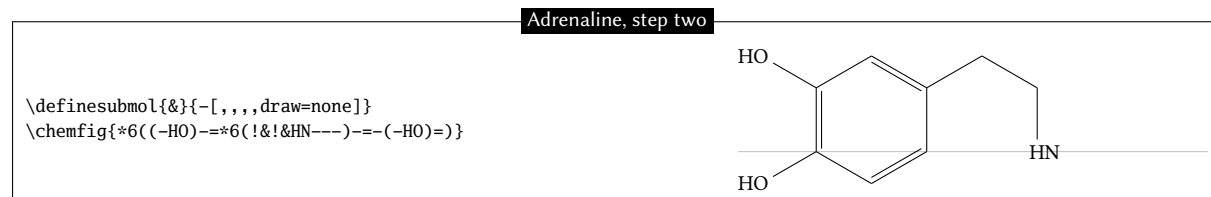
11.4.2 Using two rings

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

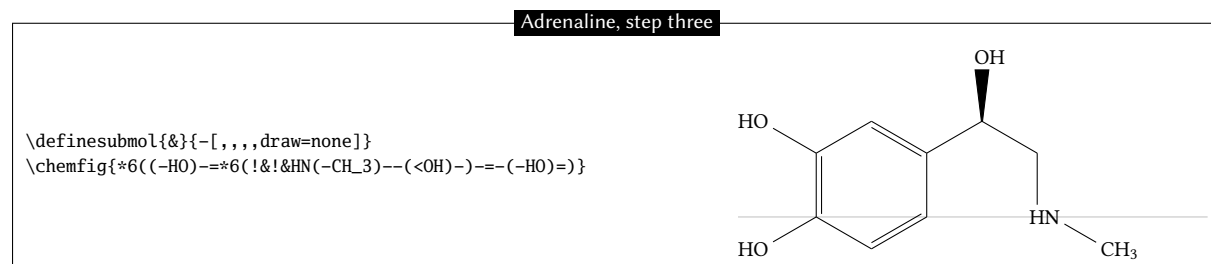
We could improve this code by considering that the drawing of the adrenaline molecule is made of two 6-rings adjacent to each other:



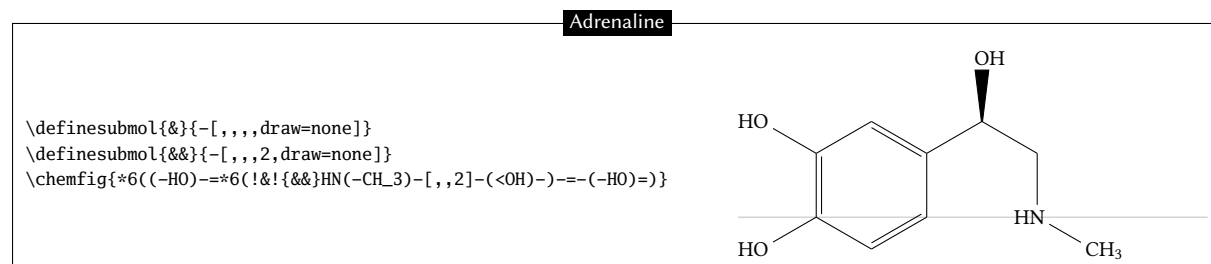
Now the first two bonds of the ring on the right need to be made invisible. To do this we use the argument that is passed to `tikz`, specifying “`draw=none`”. These bonds will thus have this code: “`-[,,,,draw=none]`”. To keep the code readable, we define an alias named “`&`” for these bonds:



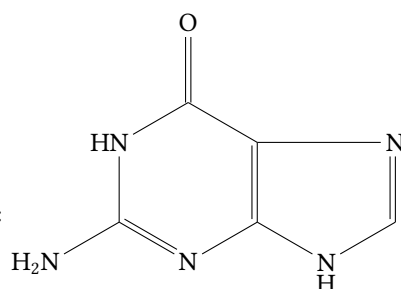
The rest becomes easy; just add the branches to the right vertices:



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`”:

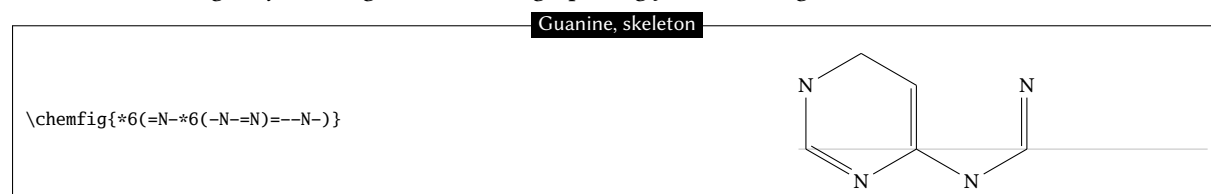


11.5 Guanine



We will draw the guanine molecule:

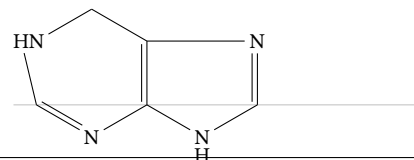
First of all, let’s begin by drawing the nested rings, putting just the nitrogen atoms at the vertices:



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:

Guanine, step two

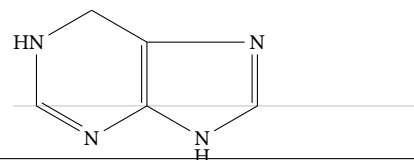
```
\chemfig{*6(=N-*6(-\chembelow{N}{H}-=N?)=?--HN-)}
```



We note that one bond leaves from the wrong atom¹⁰! 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]”:

Guanine, step three

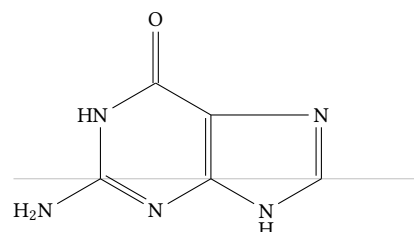
```
\chemfig{*6(=N-*6(-\chembelow{N}{H}-=N?)=?--HN-[ , , 2])}
```



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

Guanine

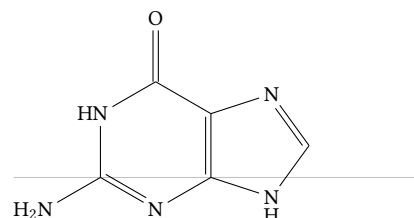
```
\chemfig{*6((-H_2N)=N-*6(-\chembelow{N}{H}-=N?)=?-(=O)-HN-[ , , 2])}
```



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.

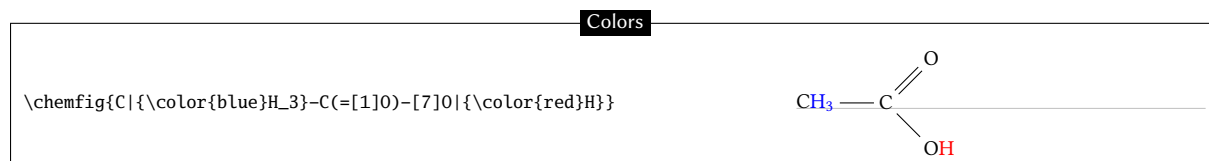
Colors

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

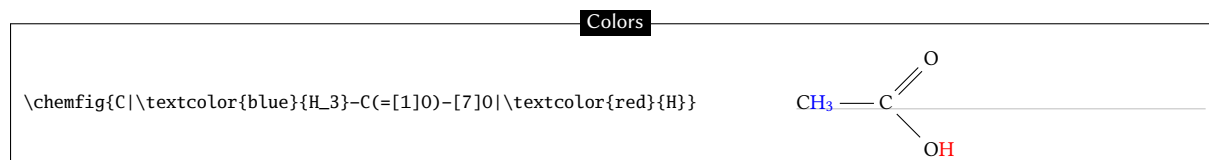


¹⁰This seems illogical because the angle of the bond from the HN group toward the first vertex lies between -90° and 90° ; `chemfig` should therefore leave from the second atom. To explain this contradiction, one must know that in rings, the last bond always links the last vertex to the first, ignoring 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.

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:



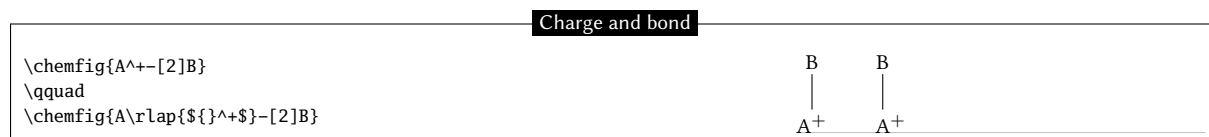
The same effect can be obtained with `\textcolor`:



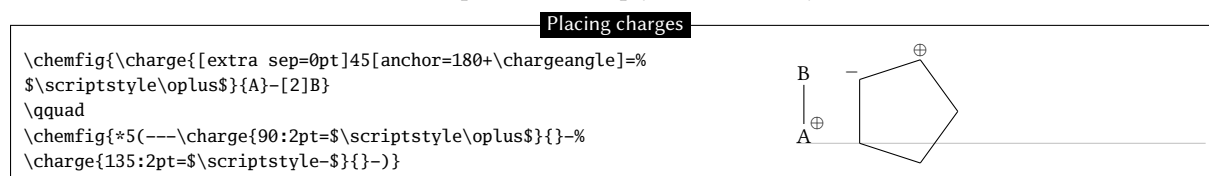
The main disadvantage is that we have to do the same for every atom that need to be colored, even if they are contiguous.

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 `\rlap`¹¹, which often gives good results. We see here that with `\rlap`, the horizontal alignment of atoms is preserved:

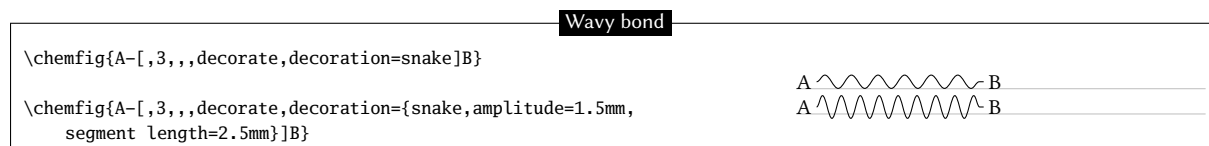


The macro `\charge` allows this task to be performed simply and accurately.



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`:

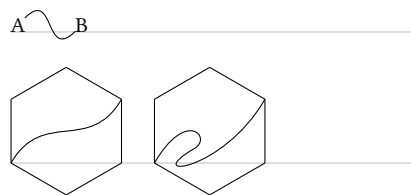
¹¹If you have to put the charge at the left of the atom, you must use the command `\llap`.

```

\chemfig{@{a}A-[,,,draw=none]@{b}B}
\chemmove{\draw[-](a)..controls +(45:7mm) and +(225:7mm)..(b);}
\bigskip

\chemfig{*6(@{a}---@{b}---)}
\chemmove{\draw[-](a)..controls +(60:3em) and +(240:3em)..(b);}
\quad
\chemfig{*6(@{a}---@{b}---)}
\chemmove{\draw[-](a)..controls +(60:3em) and +(30:1em)..
          ++(20:2em) ..controls +(210:3em) and +(-120:4em) ..(b);}

```



12.4 Draw a polymer element

The macro `\polymerdelim`, until now undocumented and in the test phase, becomes officially released in `chemfig` with version 1.33. Its syntax is as follows:

$$\backslash\text{polymerdelim}[\langle\text{keys}\rangle=\langle\text{values}\rangle]\{\langle\text{node1}\rangle\}\{\langle\text{node2}\rangle\}$$

The effect, after possibly *two* compilations, is to place vertical delimiters at the specified nodes. The parameters are specified via the `\keys` and `\values`, which are listed below, default values and actions.

<code>\keys</code>	default <code>\values</code>	Action
<code>delimiters</code>	<code>()</code>	Defines the delimiters. If these delimiters are brackets, write <code>delimiters={[]}</code> .
<code>height</code>	<code>10pt</code>	Defines the height (above the node) of the delimiters.
<code>depth</code>	<code>\vide</code>	Defines the depth (below the node) of the delimiters. If the <code>\value</code> is empty, then the depth is equal to the height.
<code>h align</code>	<code>true</code>	Boolean which, when <code>\false</code> , places the 2nd delimiter on the 2nd node, at the risk that the delimiters are not on the same horizontal line.
<code>auto rotate</code>	<code>false</code>	Boolean which, when <code>\true</code> and <code>h align = \false</code> , automatically turns the delimiters to be perpendicular to the line that connects the two nodes.
<code>rotate</code>	<code>0</code>	When <code>h align = \false</code> and <code>auto rotate = \false</code> , sets the rotation angle of the two delimiters.
<code>open xshift</code>	<code>0pt</code>	Defines the horizontal offset of the opening delimiter.
<code>close xshift</code>	<code>\vide</code>	Defines the horizontal offset of the closing delimiter. If the <code>\value</code> is empty, then this offset becomes opposite to the offset of the opening delimiter.
<code>indice</code>	<code>n</code>	Defines the indices that will be placed at the right bottom of the closing delimiter.

Polymers

```

Polyethylen:
\chemfig{\phantom{CH_2}-[@{op,.75}]CH_2-CH_2-[@{cl,0.25}]}
\polymerdelim[height = 5pt, indice = \!\!n]{op}{cl}
\bigskip

Polyvinyl chloride:
\chemfig{\phantom{CH_2}-[@{op,1}]CH_2-CH(-[6]Cl)-[@{cl,0}]}
\polymerdelim[height = 5pt, depth = 25pt, open xshift = -10pt, indice = \!\!n]{op}{cl}
\bigskip

Nylon 6:
\chemfig{\phantom{N}-[@{op,.75}]{N}(-[2]H)-C(=[2]O)-{CH_2}_5-[@{cl,0.25}]}
\polymerdelim[height = 30pt, depth = 5pt, indice = {}]{op}{cl}
\bigskip

Polycaprolactame
\chemfig[atom sep = 2em]{[:30]-[@{left,.75}]N(-[6]H)-[:30](=[2]O)-[:30]-[:30]-[@{right,0.25}:30]}
\polymerdelim[height = 5pt, indice = \!\!n]{left}{right}
\bigskip

Polyphenyl sulfide:

```

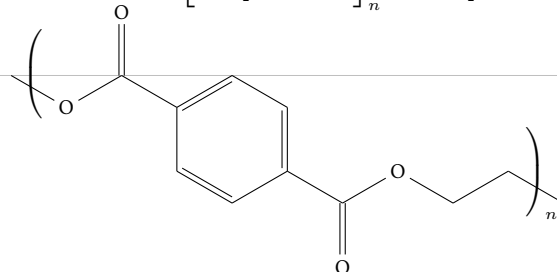
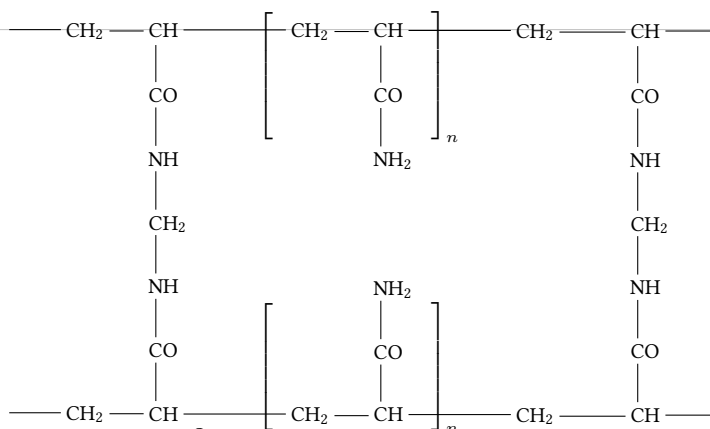
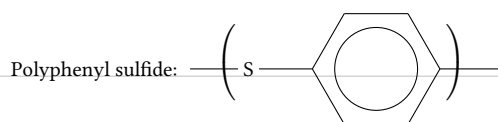
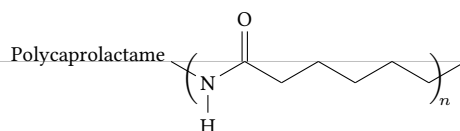
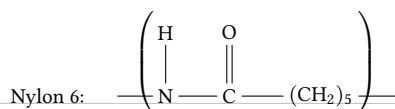
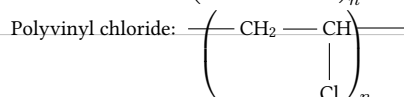
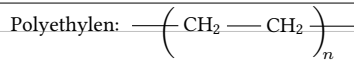
```

\chemfig{\vphantom{S}-[@{op,.75}]S-(**6(---(-[cl,0.25])---))}
\polymerdelim[delimiters = (), height = 15pt, indice = {}]{op}{cl}
\bigskip

\chemfig{-CH_2-CH([6]-CO-NH-CH_2-NH-CO-CH([4]-CH_2-)([0]-[downleft,0.8],2]CH_2
-CH([2]-CO-NH_2)-[downright,0.3],2]CH_2-[1.5]C?H-)-[upleft,0.8],2]CH_2
-CH([6]-CO-NH_2)-[upright,0.3],2]CH_2-[1.5]CH([6]-CO-NH-CH_2-NH-C?O)-}
\polymerdelim[delimiters = {[ ]}, height = 5pt, depth = 40pt, indice = n]{upleft}{upright}
\polymerdelim[delimiters = {[ ]}, height = 40pt, depth = 5pt, indice = n]{downleft}{downright}

\chemfig{-[@{op,.5}:-30]O-[:60](=[:60]O)-[:-60]*6(---(-([:-60]O)-[:60]O-[:-60]-[:60]-[cl,.5]::-60))=)}
\polymerdelim[height=6ex, indice=n, h align=false]{op}{cl}

```



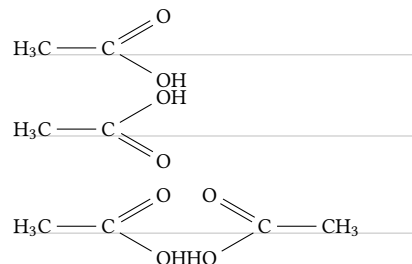
12.5 Draw the symmetrical of a molecule

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.

```

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

```



12.6 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.

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

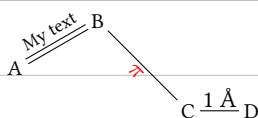
\newcommand\arcbetweennodes[3]{%
  \pgfmathanglebetweenpoints{\pgfpointanchor{#1}{center}}{\pgfpointanchor{#2}{center}}%
  \let#3\pgfmathresult}

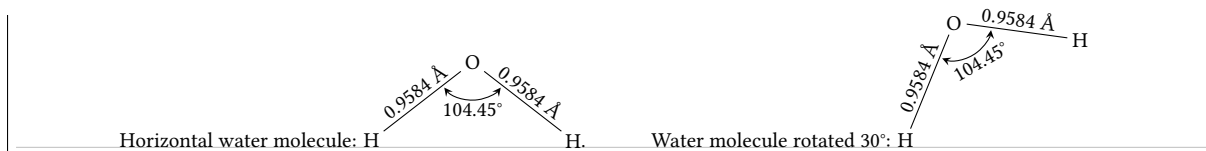
\newcommand\arclabel[6][stealth-stealth,shorten <=1pt,shorten >=1pt]{%
  \chemmove{%
    \arcbetweennodes{#4}{#3}\anglestart \arcbetweennodes{#4}{#5}\angleend
    \draw[#1]([shift=(\anglestart:#2)]#4)arc(\anglestart:\angleend:#2);
    \pgfmathparse{(\anglestart+\angleend)/2}\let\anglestart\pgfmathresult
    \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}{\small1 \angstrom}
\medskip

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

```





12.7 Dessiner des liaisons multiples

Again, the “decorations.markings” library allows to draw multiple bonds:

Liaisons multiples

```

\catcode'_{=11
\tikzset{nbond/.style args={#1}{%
  draw=none,%
  decoration={%
    markings,%
    mark=at position 0 with {\coordinate (CFstart@) at (0,0);},
    mark=at position 1 with {%
      \foreach\CF_i in {0,1,...,\number\numexpr#1-1}{%
        \pgfmathsetmacro\CF_nbondcoeff{\CF_i-0.5*(#1-1)}%
        \draw ([yshift=\CF_nbondcoeff\CF_doublesep]CFstart@)--(0,\CF_nbondcoeff\CF_doublesep);
      }%
    },
    postaction={decorate}
  }
}
\catcode'\_ =8
\chemfig{A-[1,,,nbond=4]B-[:30,,,nbond=5]C-[6,,,nbond=6]D}

```

Reaction schemes

Following several requests from users, it had become evident that `chemfig` had a weakness regarding the drawing of reaction schemes. The gap is now filled. Therefore, `chemfig` has now reached version 1.0 since I consider that the main features sought are now available.

I thank Clemens NIEDERBERGER for his help and the tests he carried out on the new features presented in this part.

1 Overview

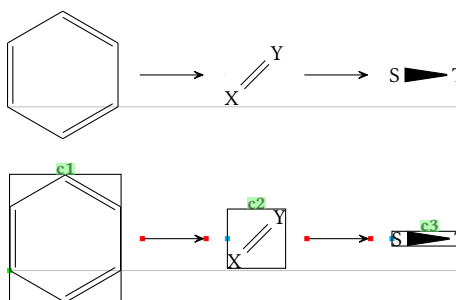
A reaction scheme must be contained between the commands “`\schemestart`” and “`\schemestop`”. As shown in this example, debug information is either hidden or displayed with the `<key>` `scheme debug` and the value `<true>` ou `<false>`:

Example 1

```

\setchemfig{scheme debug=false}
\schemestart
  \chemfig{*6(-----)}\arrow
  \chemfig{X=[1]Y}\arrow
  \chemfig{S>T}
\schemestop
\bigskip
\setchemfig{scheme debug=true}
\schemestart
  \chemfig{*6(-----)}\arrow
  \chemfig{X=[1]Y}\arrow
  \chemfig{S>T}
\schemestop

```



Some comments:

- the `\arrow` commands draw the arrows;
- everything lying between two `\arrow` commands is considered a compound. It was decided that all possible settings, whether for arrows or compounds, are controlled by the arguments of the `\arrow` command, whose syntax may become quite complex;
- arrows are plotted horizontally, this can obviously be modified;
- arrows are plotted on the imaginary line connecting the center of the compounds' bounding boxes (the red and blue squares are the anchoring points of arrows). This behavior can also be modified;
- debug information is displayed with the `scheme debug <key>`. It consists of:
 - the green label above the bounding boxes is the default name assigned to compounds by `chemfig`. It follows the series "c1", "c2", etc. Numbering is reset to 1 for every reaction scheme.
 - display of the compounds bounding boxes;
 - the arrows start and end points represented by red points and anchors by blue points;
- the distance from edge to edge between two compounds is defined with the `<key> compound sep = <dim>`. By default this `<dim>` is 5em;
- finally, the distance between the edges of the compounds and the beginning and end of the arrows is defined with the `<key> arrow offset = <dim>`. By default, this `<dim>` is 4pt.

2 Arrow types

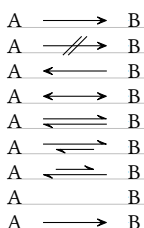
When the `\arrow` command is followed by an optional argument in braces (which is not mandatory), the argument defines the type of arrow:

Arrow types

```

\schemestart A\arrow{->}B\schemestop\par % by default
\schemestart A\arrow{-/>}B \schemestop\par
\schemestart A\arrow{<->}B \schemestop\par
\schemestart A\arrow{<->}B \schemestop\par
\schemestart A\arrow{<=>}B \schemestop\par
\schemestart A\arrow{<->>}B \schemestop\par
\schemestart A\arrow{<<->}B \schemestop\par
\schemestart A\arrow{0}B \schemestop\par
\schemestart A\arrow{-U>}B \schemestop

```



The arrow "`-U>`" is not fully drawn, an arc can be added tangent to the arrow center using optional arguments on the command, see page 57. Here is a "`-U>`" arrow with the arc on top of it: A B

For the sake of clarity, capital letters will be used throughout the documentation instead of chemical formulas made with the `\chemfig` command except for specific examples. Reaction schemes obviously work identically with letters and drawn molecules. Several examples are shown in the Gallery with proper reaction schemes.

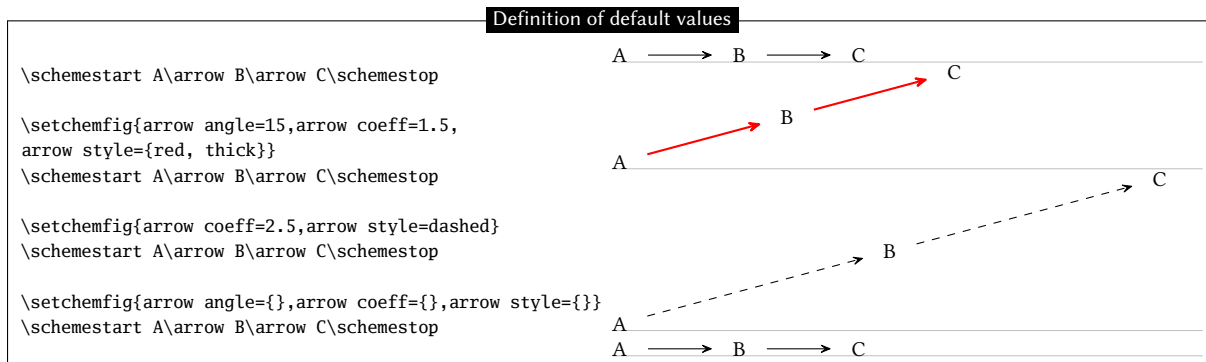
3 Arrows features

Each arrow is characterized by:

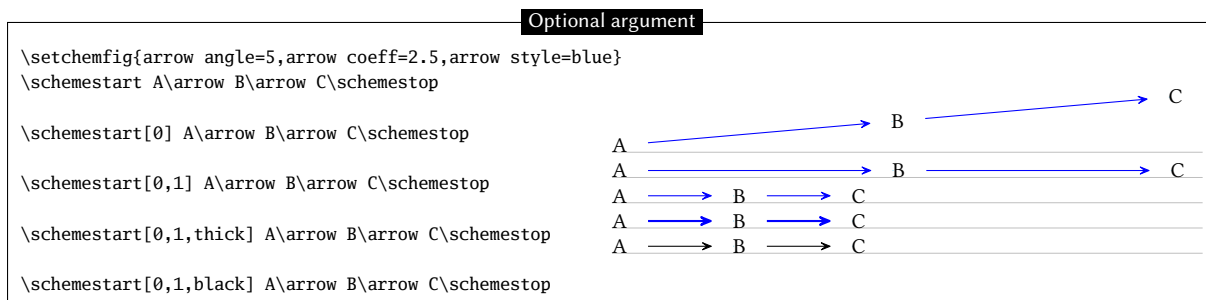
- an angle expressed in degrees;
- a coefficient that specifies the arrow length through the multiplication of the compounds spacing value defined by `compound sep`;
- a style with `tikz` instructions to customize the color, the thickness or other graphical attribute of the arrow.

These features are defined with the `<keys>`

- `arrow angle = <angle>`, which default value is 0;
- `arrow coeff = <decimal>`, which default value is 1;
- `arrow style = <code tikz>`, empty by default.

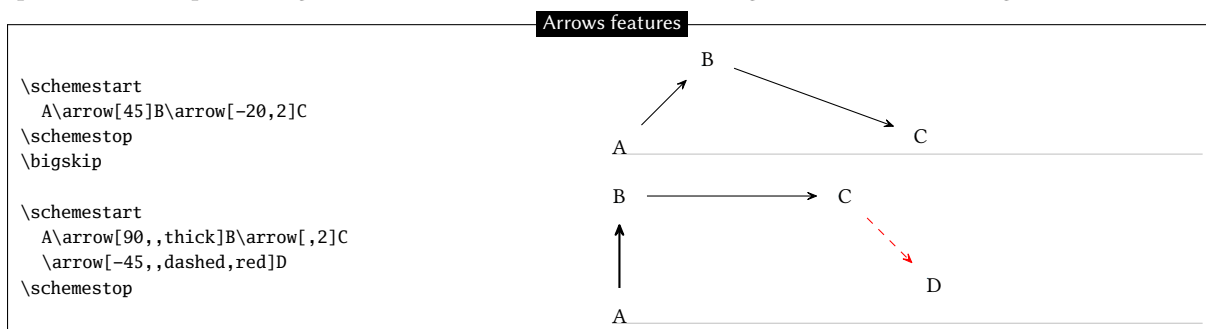


In order to locally modify one or all of these default values, the `\schemestart` command accepts an optional argument in the form `[angle,coeff,style]` which changes the default arrow features within the sole reaction scheme:



Regarding style, the rule is: the style specified in the argument in brackets applies *after* the default style, without overwriting it! This is why only the “black” color attribute is able to overwrite the “blue” default style.

Finally, the `\arrow` command accepts an optional argument in brackets in the form `[angle,coeff,style]` to change the feature of that given arrow. As above, style applies *after* the default style and *after* the style possibly-specified in the optional argument of the `\schemestart` command, again without overwriting them.



4 Compounds names

Automatic naming of compounds (“c1”, “c2”, etc.) can be overridden. For this, the `\arrow` command must be immediately followed by an argument in parentheses. The argument is of the form: `(n1--n2)`. The compounds located at the beginning and at the end of the arrow are named “n1” and “n2”, respectively. Any alphanumeric

string can be used. The numbering of the names "c<n>" continues internally, so if a compound has a different name than the default one, it does not affect the default name of the subsequent compounds.

Names are optional, and the argument can be either (n1--) and (--n2).

Compounds names

```

\setchemfig{scheme debug=true}
\schemestart
  A\arrow(aa--bb)B\arrow(--cc)C\arrow(--dd)D\arrow E
\schemestop
\bigskip
\schemestart
  A\arrow(aa--)B\arrow(bb--)C\arrow(cc--dd)D\arrow E
\schemestop

```

Note that both methods are equivalent. Therefore, compounds can either be named by arrows preceding or following them. However, when a compound is surrounded by two arrows specifying its name, the first name is ignored and a warning message is generated:

Overfull naming

```

\setchemfig{scheme debug=true}
\schemestart
  A\arrow(--foo)B\arrow(bar--)C
\schemestop

```

Here compound "B" is called "foo" by the arrow pointing at it, and "bar" by the arrowing leaving from it. Thus chemfig generates a warning mentioning that the name "foo" will be ignored:

Package chemfig Warning: two names for the same node, first name "foo" ignored

5 Anchoring

As noted above, arrows lie on the line connecting the center of the compounds' bounding boxes. Default anchors are called "center" in the sense of tikz. Non-default anchoring points can be user-specified as well with an argument between brackets:

(n1. a1--n2. a2)

where the anchor "a1" or "a2" can be: north west, north, north east, west, center, east, mid west, mid, mid east, base west, base, base east, south west, south, south east, text, or any angle. Here is an example from the tikz manual where the anchors are located on the bounding box:

TikZ anchoring

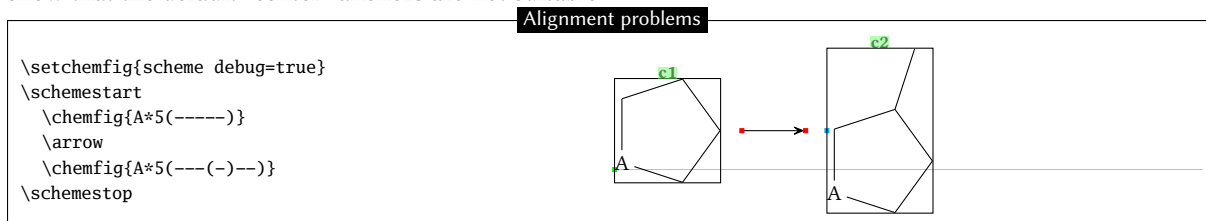
```

\Huge
\begin{tikzpicture}[baseline]
\node[anchor=base west,name=x,draw,inner sep=25pt] {\color{lightgray}Rectangle\vrule width 1pt height 2cm};
\foreach \anchor/\placement in
{north west/above left, north/above, north east/above right,west/left, center/above, east/right,
mid west/left, mid/above, mid east/right,base west/left, base/below, base east/right,
south west/below left, south/below, south east/below right,text/below,10/right,45/above,150/left}
\draw[shift=(x.\anchor)] plot[mark=x] coordinates{(0,0)}
\node[\placement,inner sep=0pt,outer sep=2pt] {\scriptsize\texttt{(\anchor)}};
\end{tikzpicture}

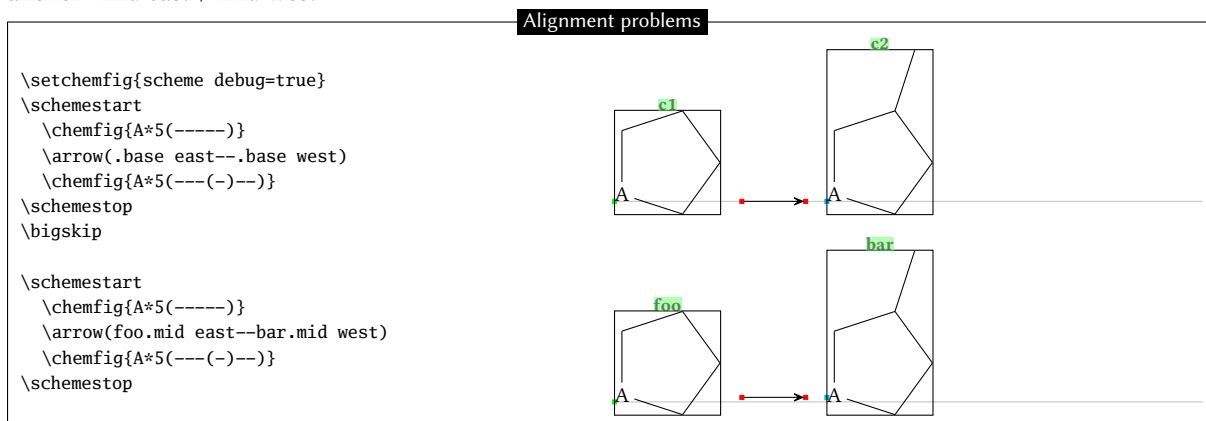
```

Like for names, arrival and departure anchoring points are independent and optional.

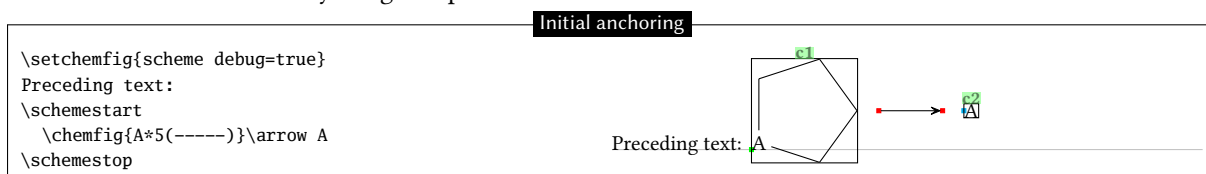
In this example, the default alignment is not good because the two “A” are not aligned vertically. Debug information show that the default “center” anchors are not suitable:



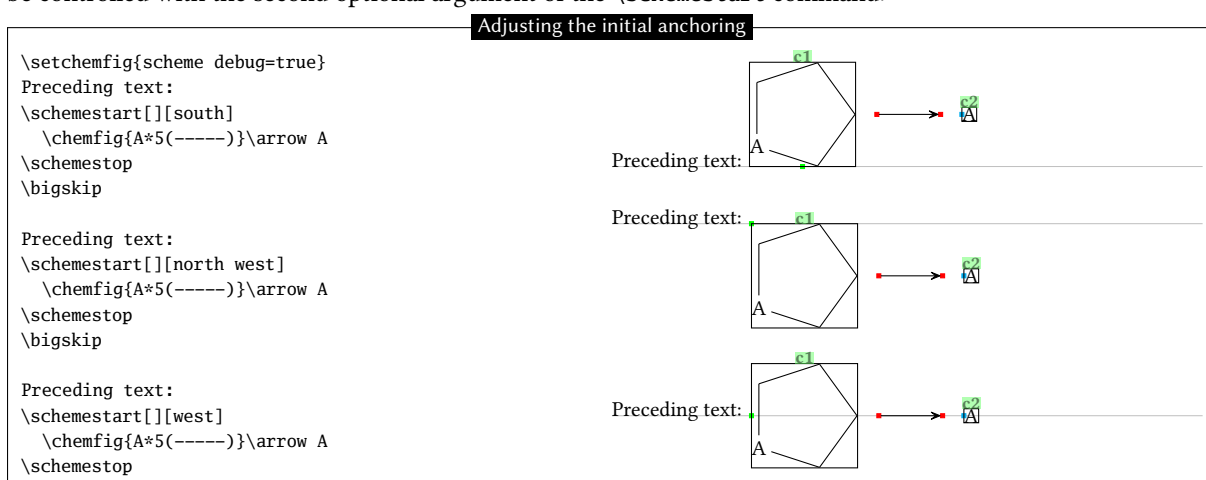
For the alignment to be correct, arrows will leave/arrive either from the anchor “base east”/“base west”, or from anchor “mid east”/“mid west”:



One last anchor need be specified: the anchor of the first compound with respect to the baseLine of the text just before it. This is illustrated by the green point on the left-hand side of the scheme below:



The default position of this anchor on the first compound’s bounding box is that given by “text”. This position can be controlled with the second optional argument of the \schemestart command:



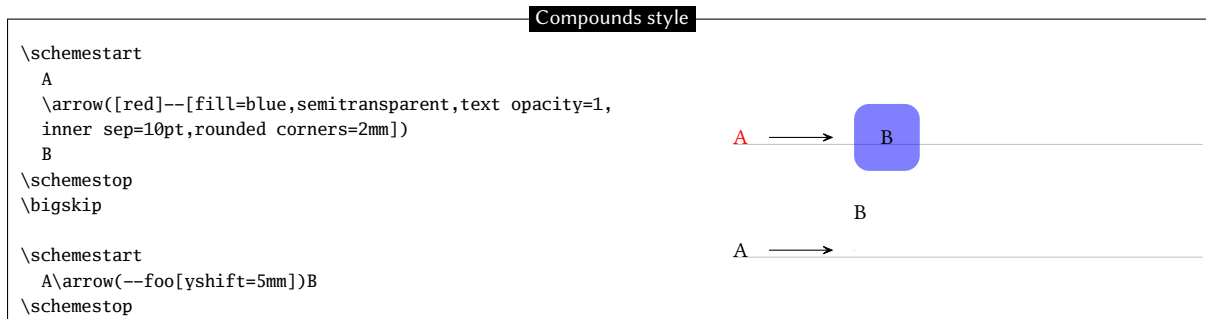
6 Compounds style

The \arrow command can also include tikz instructions to define the bounding box style “s” of the reactant and the product of the reaction. This is done with the argument between parentheses. Always style through the argument

in brackets of the `\arrow`, we can specify with `tikz` instructions the style “s” to bounding box of the compound of departure or of arrival. Therefore the complete syntax of the `\arrow` command, with each specification being optional, is as follows:

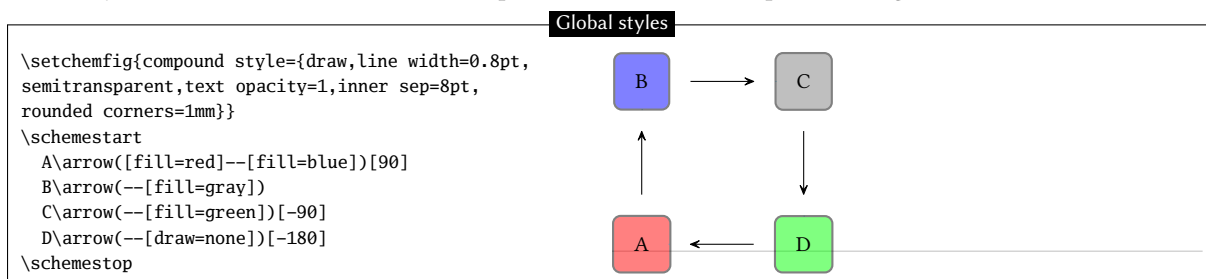
$$\backslash\text{arrow}(n1.a1[s1]--n2.a2[s2])\{\text{arrow type}\}[\text{angle},\text{coeff},\text{arrow style}]$$

Like for names, if specific styles are given to one compound by arrows arriving on it and leaving from it, the first style will be ignored with a warning.



The macro `\setcompoundstyle{<code tikz>}` allows to globally define the style of compounds displayed thereafter. Entering an empty argument results in the absence of style, which corresponds to the default case.

Here a style is defined with round corner-shaped boxes and semitransparent background:

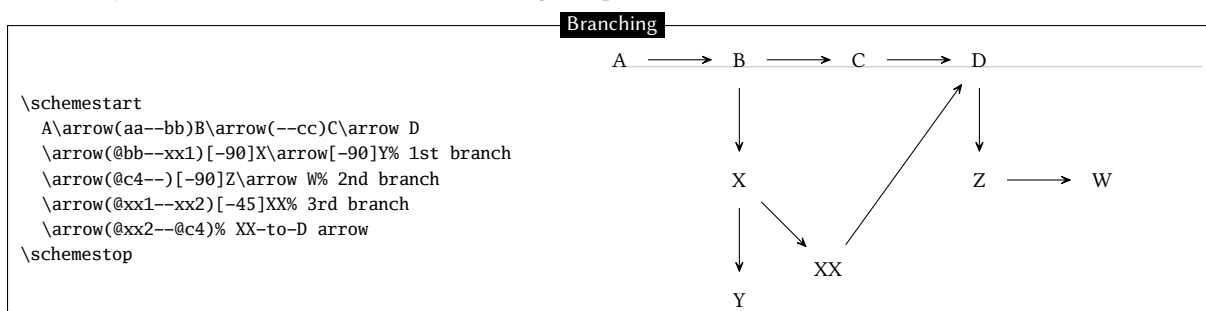


7 Branching

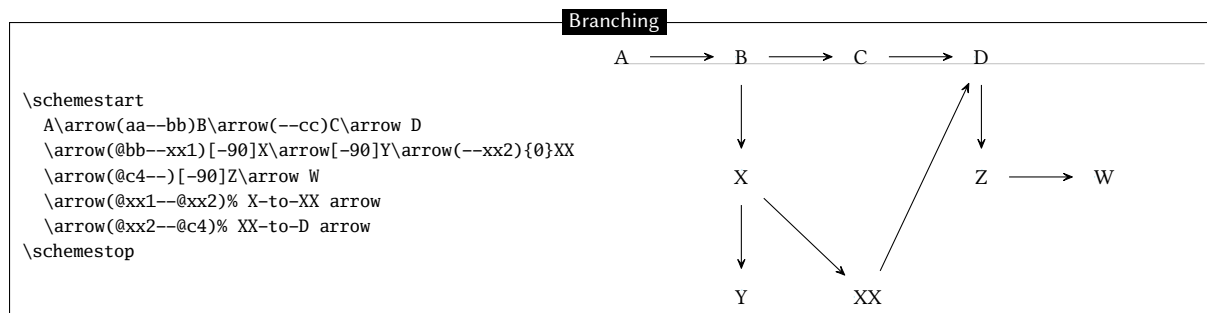
So far, only linear reaction schemes have been treated. Branched schemes are also possible and this is where compound names play a key role. When a name is preceded by “@” in the argument between brackets of the `\arrow` command, it means that the compound already exists. Several scenarios are possible:

- (`@n1--n2`): the arrow will leave from the existing compound “n1” and the scheme will continue following the arrow, thus creating a branch;
- (`@n1--@n2`): the arrow is drawn between two existing compounds, no matter whether they are already defined or whether they will later in the reaction scheme: therefore this syntax can be placed *anywhere* in the code of the reaction scheme;
- (`n1--@n2`): this syntax is not permitted;

In the following example, 3 branches are made, a first one from “B”, a second one from “D” and a last one from “X”. Finally one more arrow connects two existing compounds: “XX” and “D”:

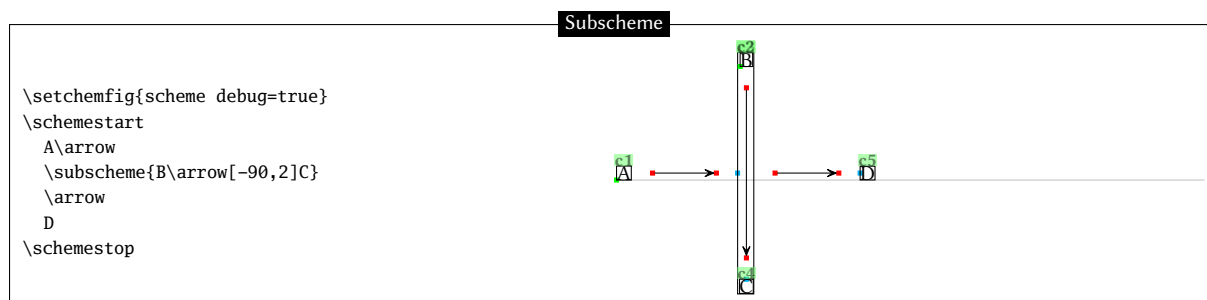


One may wish to have “Y” and “XX” on the same horizontal line. To achieve this, a horizontal invisible bond is drawn between “Y” and “XX”; the scheme is completed with a final arrow between the two existing compounds “XX” and “D”:

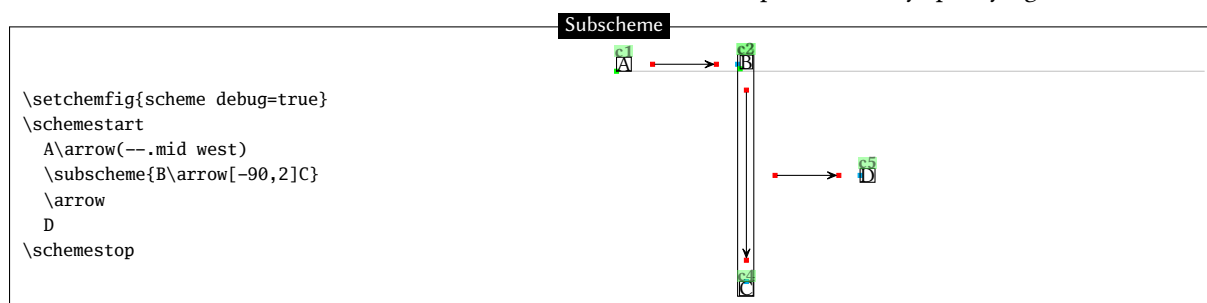


8 Subscheme

A fraction of the reaction scheme can be defined within a single bounding box, so that `chemfig` treats it as a compound. The reaction scheme fraction is defined inside the compulsory argument between braces of the `\subscheme` command so it is subsequently regarded as a single entity. When `\subscheme` is located after an arrow, the command labels this subscheme as a compound named “`c<n+1>`”:



Although this is not clearly seen because of labels overlap, the box around the subscheme is called “`c2`”, and name numbering continues inside the subscheme with B called “`c3`” and C called “`c4`”. Since the first compound in the subscheme is “B”, the subscheme’s baseline is that of “B”. This can be pointed out by specifying the anchors:



Note that since “`\subscheme{<scheme>}`” is only a convenient shortcut for

```
\schemestart<scheme>\schemestop
```

Consequently, it can be used with the same optional arguments as `\schemestart`.

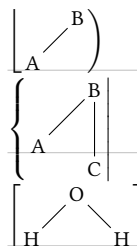
`chemfig` provides the `\chemleft` and `\chemright` command pair. These allow to set expandable delimiters on either side of a material. The commands must be followed by delimiters, just like in the case of \TeX primitive commands `\left` and `\right`:

```
\chemleft<car1><material>\chemright<car2>
```

where `<car1>` and `<car2>` can be “(” et “)” or “[” and “]”, or any other expandable delimiter consistent with the `\left` et `\right` commands.

The `\chemleft` and `\chemright` macros

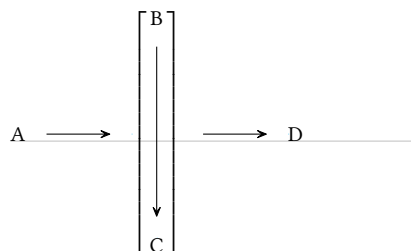
```
\chemleft\lfloor\chemfig{A-[1]B}\chemright)
\chemleft\{\chemfig{A-[1,1.25]B-[6,1.25]C}\chemright|
\chemleft[\chemfig{H-[1]O-[7]H}\chemright]
```



The code of the reaction scheme discussed above including `\chemleft` and `\chemright` is written:

Reaction scheme with `\chemleft` and `\chemright`

```
\schemestart
A\arrow
\chemleft[\subscheme{B\arrow[-90,2]C}\chemright]
\arrow
D
\schemestop
```



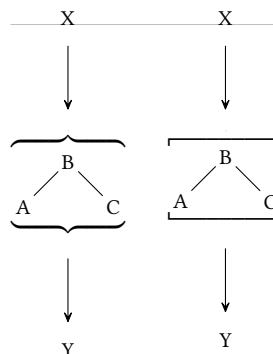
By analogy, the macros `\chemup` and `\chemdown` can be used to draw expandable delimiters above and below the material, respectively:

`\chemup<car1>\chemdown<car2>`

For example:

The `\chemup` and `\chemdown` macros

```
\schemestart[-90]
X\arrow
\chemup\{\chemfig{A-[1]B-[7]C}\chemdown\}
\arrow Y
\schemestop
\qqquad
\schemestart[-90]
X\arrow
\chemup[\chemfig{A-[1]B-[7]C}\chemdown]
\arrow Y
\schemestop
```



Delimiters can also be drawn through compounds' style and apply them to a random compound (and hereby to a subscheme). These expandable delimiters (parentheses, brackets, braces) can be used upon loading the "matrix" tikz library in the document preamble:

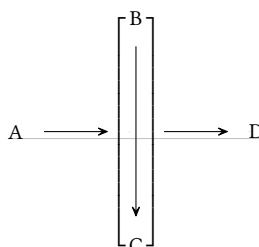
`\usetikzlibrary{matrix}`

Since the `\chemleft`, `\chemright`, `\chemup` and `\chemdown` commands are available, the `chemfig` package will *not* automatically load the library. As long as the user want to access this special set of delimiters, the library must be explicitly loaded.

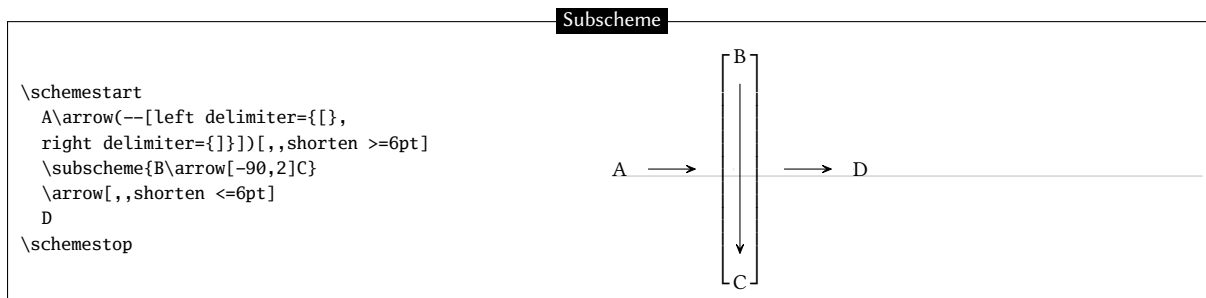
The same brackets-delimited subscheme as above is presented again:

The "matrix" library delimiters

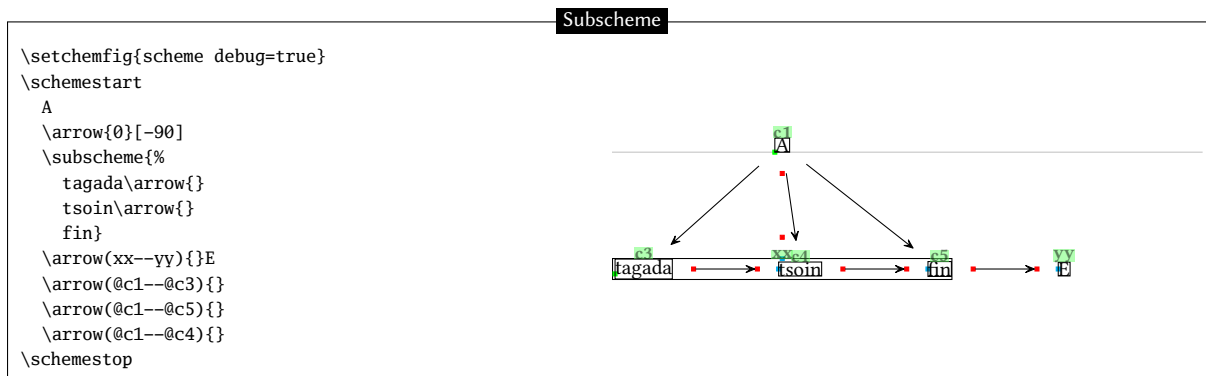
```
\schemestart
A\arrow(--[left delimiter={[]}, right delimiter={}])
\subscheme{B\arrow[-90,2]C}
\arrow
D
\schemestop
```



Since the delimiters are drawn outside the bounding box, it is advisable to slightly shorten the incoming and outgoing arrows:

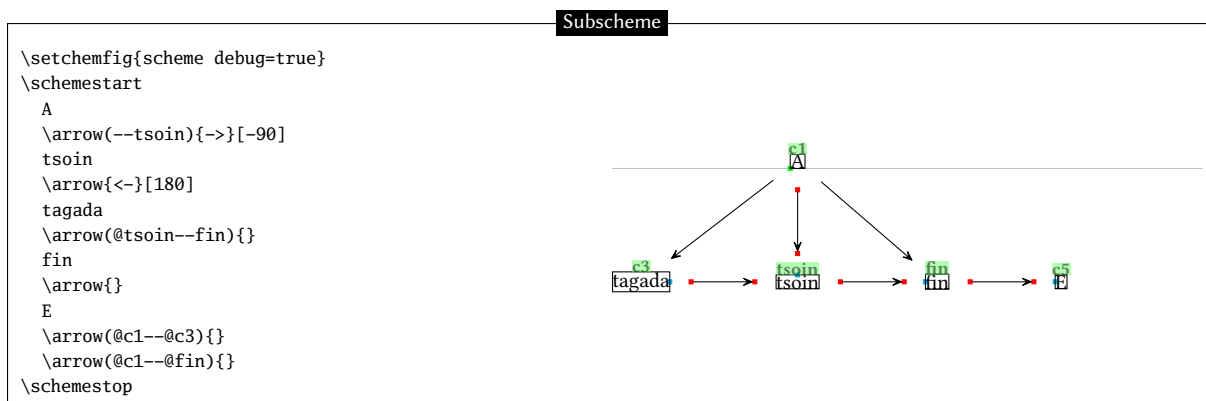


Subschemes should be used with care, undesired results are sometimes observed. In this example, a subscheme is used to horizontally align 3 different compounds:



The center of the subscheme is exactly located on the same vertical line as the center of compound "A". This is because the two entities are connected by an invisible arrow with a -90 angle. However, the arrow between the two pre-existing compounds "A" and "tsoin" is *not* vertical because "tsoin" is not on the center of the subscheme since "tagada" is wider than "end". If this arrow is to be vertical within the use of the `\subscheme` command, one must find a correct angle for the arrival anchor of the invisible arrow by try-and-error.

A much simpler method is to use a branch instead of a subscheme: draw a *visible* arrow between "A" and "tsoin", and then draw horizontal arrows on both sides of "tsoin", with a branch for the right-hand side arrows.



9 Arrows optional arguments

Within the argument in braces of the `\arrowcommand`, the arrow name can be followed by optional arguments written between brackets. Here are the possible values for these optional arguments and their meaning, as defined by `chemfig`:

- the arrows " \rightarrow ", " \leftarrow ", " \leftrightarrow ", " \Leftrightarrow ", " \Leftrightarrow ", " \Leftrightarrow ", " \rightarrow " have three optional arguments:
 - the first one contains the "label" to be placed above the arrow;

- the second one contains the “label” to be placed below the arrow. The orientation of these two labels is given by the same angle as the arrow. The perpendicular shift between the arrow and the label anchor can be adjusted with the `<key> arrow label sep = <dim>` which value is 3pt by default. Labels contained in the two optional arguments are *not* typed in math mode.
- the third one is a dimension corresponding to a shift perpendicular to the arrow that can be applied to it: the dimension is positive for an upward shift of the arrow (and of its labels, if any), and negative for a downward shift.
- the “-U>” arrow has 5 optional arguments:
 - the first three are identical to those found in the other arrow types;
 - the fourth one is a coefficient (which is 0.333 by default) which multiplies the length of the arrow to get the radius of the arc;
 - the fifth one is the half-angle from the center of the arc path, it is 60 degrees by default.
- the invisible arrow “0” accepts two optional arguments of the same type as the first two listed above;

Arrows optional arguments

```

\setchemfig{scheme debug=false}
\schemestart A\arrow{->[up][down]}B \schemestop
\qqquad
\schemestart A\arrow{->[up][down][4pt]}B \schemestop
\qqquad
\schemestart A\arrow{->[up][down][-4pt]}B \schemestop
\medskip
\schemestart A\arrow{<=>[up][down]}{30,1.5}B \schemestop
\medskip
\schemestart[-20]
A\arrow{->}B\arrow{->[[]][3pt]}C\arrow{->[[]][-3pt]}D
\schemestop

```

A problem arises for vertical arrows:

Vertical arrows

```

\schemestart
A\arrow{->[up][down]}[-90]B
\schemestop

```

For the sake of clarity, one may prefer to have the “above” and “below” labels written horizontally. Label angles can be specified, while default is the same angle as that of the arrow. To choose a specific angle, `*{<angle>}` can be written at the beginning of the optional arguments:

Choice of angles

```

\setchemfig{scheme debug=true}
\schemestart A\arrow{->[*{0}up][*{0}down]}[90]B\schemestop
\qqquad
\schemestart A\arrow{->[*{0}up][*{0}down]}[45]B\schemestop
\qqquad
\schemestart A\arrow{->[*{0}up][*{0}down]}[-45]B\schemestop
\qqquad
\schemestart A\arrow{->[*{0}up][*{0}down]}[-90]B\schemestop

```

The default position of the label anchor can lead to undesired results:

Anchors

```

\setchemfig{scheme debug=true}
\schemestart
A\arrow{->[*{0}on top of][*{0}underneath]}[45,2]B
\schemestop

```

To counter this, the anchoring position can be specified as well to override the one selected by chemfig by default. The syntax for this is: `*{⟨angle⟩}.⟨ancre⟩}`.

Anchors

```

\setchemfig{scheme debug=true}
\schemestart
  A\arrow{->[*{0.0}on top of][*{0.180}underneath]}{45,2}B
\schemestop
\quad
\schemestart
  A\arrow{->[*{0.south east}on top of][*{0.north west}underneath]}{45,2}B
\schemestop

```

The “-U>” arrow remains a particular case. If one of the two labels from the first two optional arguments is present, the corresponding arc is plotted:

The -U> arrow

```

\schemestart A\arrow{-U>[123][456]}B\schemestop
\quad
\schemestart A\arrow{-U>[123]}{30}B\schemestop
\quad
\schemestart A\arrow{-U>[456]}{-30}B\schemestop

```

The fourth and fifth optional arguments modify the appearance of the arc: respectively the arrow length coefficient which sets the arc radius, and the angle that defines the half arc:

The -U> arrow

```

\schemestart A\arrow{-U>[123][456][][0.25]}B\schemestop
\quad
\schemestart A\arrow{-U>[123][456][][90]}B\schemestop
\quad
\schemestart A\arrow{-U>[123][456][1][45]}B\schemestop

```

With negative values for the radius and the angle, the arc is drawn below the arrow:

The -U> arrow

```

\schemestart
  A\arrow{-U>[123][456][][-0.333][-60]}B
\schemestop

```

Label angles and anchoring customization is controlled with the first two arguments, just like for other arrows:

The -U> arrow

```

\schemestart
  A\arrow{-U>[123][456]}{-90}B
\schemestop
\quad
\schemestart
  A\arrow{-U>[*{0.180}123][*{0.180}456]}{-90}B
\schemestop

```

10 Arrows customization

This section is quite technical and requires some knowledge of `tikz`. It is targeted at advanced users only who need to define their own arrows.

The `\definearrow` command allows to build custom arrows. Its syntax is:

$$\definearrow{\langle number \rangle}{\langle arrow name \rangle}{\langle code \rangle}$$

where `⟨number⟩` is the number of optional arguments that will be used in the `⟨code⟩`, with the usual syntax #1, #2, etc. These optional arguments cannot accept default values; if no value is specified upon using the macro `\arrow`, the arguments will remain empty.

Before going further, let's examine the available internal macros when drawing arrows. Since these macros include the “@” character in their name, they can only be accessed between `\catcode'_ =11` and `\catcode'_ =8` commands.

- `\CF_arrowstartname` and `\CF_arrowendname` include the names of the compounds (considered as nodes by `tikz`) between which the arrow is drawn;
- `\CF_arrowstartnode` and `\CF_arrowendnode` include the node names where arrow ends will be located. After these names, user-defined anchors can be specified in the argument between brackets of the `\arrowcommand`, unless the field is left empty;
- `\CF_arrowcurrentstyle` and `\CF_arrowcurrentangle` contain the style and the angle of the arrow to be drawn;
- `\CF_arrowshiftnodes{<dim>}` shifts the nodes “`\CF_arrowstartnode`” and “`\CF_arrowendnode`” perpendicularly relative to the arrow by a dimension specified in the argument;
- `\CF_arrowdisplaylabel{#1}{#2}{#3}{#4}{#5}{#6}{#7}{#8}` is the most complex one. It gives the labels position with the following arguments:
 - #1 and #5 are the labels to be written;
 - #2 and #6 are real numbers between 0 and 1. They specify the location of the labels on the arrow. 0 is the beginning of the arrow and 1 is its end, assuming a *straight* arrow;
 - #3 and #7 are the “+” or “-” characters. “+” displays the label above the arrow, while “-” does it below it;
 - #4 and #8 are the names of the nodes corresponding to the beginning and the end of the arrow.
- arrow heads are based on “CF” for a full arrow and have the “harpoon” option for half arrows.

10.1 First arrow

As an example, assume we want to make an arrow with a circle on its center. Let’s call it “`-.>`”. This arrow will accept four optional arguments. Like for previously-defined arrows, the first and second arguments will be the labels to be located above and below the arrow. The third one will define the perpendicular shift relative to the arrow direction. Finally, the 4th argument will define the circle size. If this last argument is absent the default circle size will be equal to 2pt.

Let’s start with `\definearrow{4}{-.>}` to declare that the arrow will have 4 optional arguments and that it will be called `-.>`. First, the position of the nodes between which the arrow is to be drawn must be modified in order to take the third-argument shift into account. This is made with the macro `\CF_arrowshiftnodes`, so the code of the arrow will start with: `\CF_arrowshiftnodes{#3}%`. Then, one must plot the arrow itself, while taking the opportunity to set a node on the center of the segment, which will be called “`mid@point`”. Finally, the circle is defined with its center on that node. The whole `tikz` code is:

```
\edef\pt_radius{\ifx\empty#4\empty 2pt\else #4\fi}% circle radius
\expandafter\draw\expandafter[\CF_arrowcurrentstyle,-CF]
(\CF_arrowstartnode)--(\CF_arrowendnode)coordinate[midway](mid@point);
\filldraw(mid@point)circle(\pt_radius);%
```

The last step is to enter the labels, if any, with the following line:

```
\CF_arrowdisplaylabel{#1}{0.5}{+}{\CF_arrowstartnode}{#2}{0.5}{-}{\CF_arrowendnode}
```

Here is the completed arrow:

Arrow “-.>”

```
\catcode'\_11
\definearrow4{-.>}{%
\edef\pt_radius{\ifx\empty#4\empty 2pt\else #4\fi}% dot radius
\CF_arrowshiftnodes{#3}%
\expandafter\draw\expandafter[\CF_arrowcurrentstyle,-CF](\CF_arrowstartnode)--(\CF_arrowendnode)
coordinate[midway](mid@point);
\filldraw(mid@point)circle(\pt_radius);%
\CF_arrowdisplaylabel{#1}{0.5}{+}{\CF_arrowstartnode}{#2}{0.5}{-}{\CF_arrowendnode}
}
\catcode'\_8
\schemestart
A \arrow{-.>} B \arrow{-.>[above][below][][1pt]} C \arrow{-.>[below]}{30} D \arrow{-.>[above][][5pt][1.5pt]} E
\schemestop
```

10.2 Curved arrow

How about a curved arrow? To make things as simple as possible, assume it will have one single optional argument with the `tikz` code that will specify the point(s) of control. If this argument is empty, a “-CF” type arrow will be plotted.

If #1 is not empty, attention should not be drawn to “\CF_arrowstartnode” and “\CF_arrowendnode” which contain the node names of arrow ends positions, because the location of these nodes is already determined by the anchors calculated for *straight* arrows! Instead we will use \CF_arrowstartname and \CF_arrowendname which contain the names of the compound (which are nodes for *tikz*), since the arrow must be plotted between them. Here’s the *tikz* code to draw the curved arrow between the two compounds:

```
\draw[shorten <=\CF_arrowoffset,shorten >=\CF_arrowoffset,\CF_arrowcurrentstyle,-CF,
(\CF_arrowstartname).. controls #1 ..(\CF_arrowendname)];%
```

One must add a *tikz* code to shorten the arrow by an amount \CF_arrowoffset defined by \setarrowoffset. Indeed, the nodes are not the same as those for straight arrows (\CF_arrowstartnode and \CF_arrowendnode). So before \CF_arrowcurrentstyle, the following code must be added:

```
shorten <=\CF_arrowoffset, shorten >=\CF_arrowoffset
```

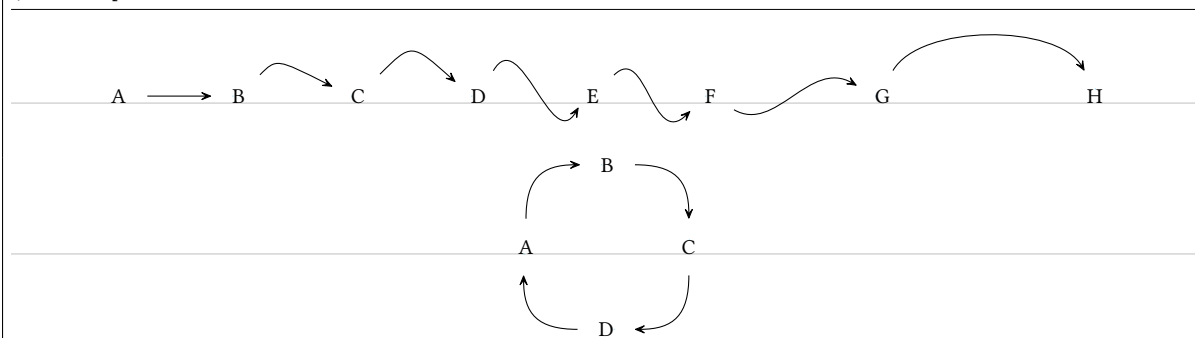
this is the role the two lines after \else.

So here is our curved arrow:

Curved arrow

```
\catcode'\_11
\definearrow1{s>}{%
\ifx\empty#1\empty
\expandafter\draw\expandafter[\CF_arrowcurrentstyle,-CF](\CF_arrowstartnode)--(\CF_arrowendnode);%
\else
\def\curvedarrow_style{shorten <=\CF_arrowoffset,shorten >=\CF_arrowoffset,}%
\CF_eaddtomacro\curvedarrow_style\CF_arrowcurrentstyle
\expandafter\draw\expandafter[\curvedarrow_style,-CF](\CF_arrowstartname)..controls#1..(\CF_arrowendname);
\fi
}
\catcode'\_8
\schemestart
A\arrow{s>}
B\arrow{s>[+(0.5cm,0.5cm)]}
C\arrow{s>[+(45:1cm)]}
D\arrow{(.60--.120){s>[+(60:1cm) and +(-120:1cm)]}
E\arrow{s>[+(45:1) and +(-135:1)]}
F\arrow{s>[+(-30:1) and +(150:1)]}[1.5]
G\arrow{(.90--.90){s>[+(60:1)and+(120:1)]}[2]
H
\schemestop

\schemestart
A\arrow{(.90--.180){s>[+(90:0.8) and +(180:0.8)]}[45]B
\arrow{(.0--.90){s>[+(0:0.8) and +(90:0.8)]}[-45]C
\arrow{(.90--.0){s>[+(-90:0.8) and +(0:0.8)]}[-135]D
\arrow{(.180--.90){s>[+(180:0.8) and +(-90:0.8)]}[135]
\schemestop
```



11 The `\merge` command

The `\merge` command allows to draw arrows coming from several existing compounds that merge into one single arrow that arrive to one single compounds.

Just after the `\merge` command, the direction that follows up must be specified. For this, 4 different direction characters can be used: “>” (the default direction if no character is entered), “<”, “^” and “v”.

The syntax follows with:

$$\backslash\text{merge}\{\text{dir}\}(n1.a1)(n2.a2)(\dots)(ni.ai)--(n.a[s])$$

where the “ni” names before the double dash are those already-defined compounds from which out coming arrows will merge into a single one. One can also specify the “ai” anchor, when the default one is not convenient. Like for the `\arrowcommand`, the command “n.a[s]” includes the name, the anchor and the style of the target compound.

The `\merge` command

```

\schemestart
ABC\arrow[30]EFGHIJ\arrow[45]KLM\arrow[60]NO
\merge>(c1)(c2)(c3)--()series 1
\arrow series 2
\schemestop
\bigskip

\schemestart
Fofoo\arrow(fooo--bar){<=>}Bar\arrow(--baz){<=>}Bz
\merge^(foo)(bar)(baz)--()series
\schemestop
\bigskip

\setchemfig{scheme debug=true}
\schemestart
A\arrow{<->}[90]B
\merge<(c1.120)(c2)--(foobar.45[circle,blue])CCC
\schemestop

```

Regarding the geometry of the `\merge` arrow, it consists of n segments leaving from n compounds up to the perpendicular line that connects them: the default length of the shortest of these segments is worth half of the compound-spacing distance defined by `\setcompoundsep`. The arrow drawn from the connecting line to the target compound has the same default length, its origin is on the middle of the connecting line. These three geometric features can be customized with the optional argument immediately after the target compound:

$$\backslash\text{merge}\{\text{dir}\}(n1.a1)(n2.a2)(\dots)(ni.ai)--(n.a[s])[c1,c2,c,\text{style}]$$

where:

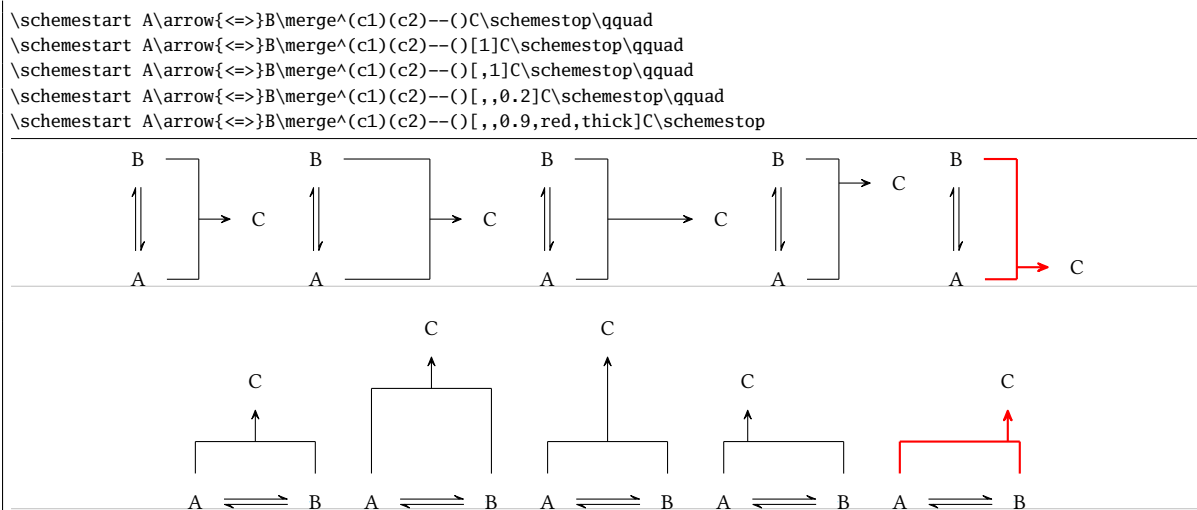
- the shortest segment distance between reactants and the connecting line is controlled through the multiplication of the `\setcompoundsep` distance by a coefficient `c1`, whose default value is 0.5;
- the length of the arrow between the connecting line and the product compound is controlled through the multiplication of the `\setcompoundsep` distance by a coefficient `c2`, whose default value is 0.5;
- the origin of the arrow between the connecting line and the product compound is determined by the coefficient `c`, a value of 0 involves a departure from the the left of the connect line (or from its top if the direction is v or ^);
- the style of the `\merge` arrow is defined with the last argument: `style`.

Geometrical parameters of `\merge`

```

\schemestart A\arrow{<=>}[90]B\merge(c1)(c2)--()C\schemestop\quad
\schemestart A\arrow{<=>}[90]B\merge(c1)(c2)--()[1]C\schemestop\quad
\schemestart A\arrow{<=>}[90]B\merge(c1)(c2)--()[,1]C\schemestop\quad
\schemestart A\arrow{<=>}[90]B\merge(c1)(c2)--()[,0.2]C\schemestop\quad
\schemestart A\arrow{<=>}[90]B\merge(c1)(c2)--()[,0.9,red,thick]C\schemestop
\bigskip

```

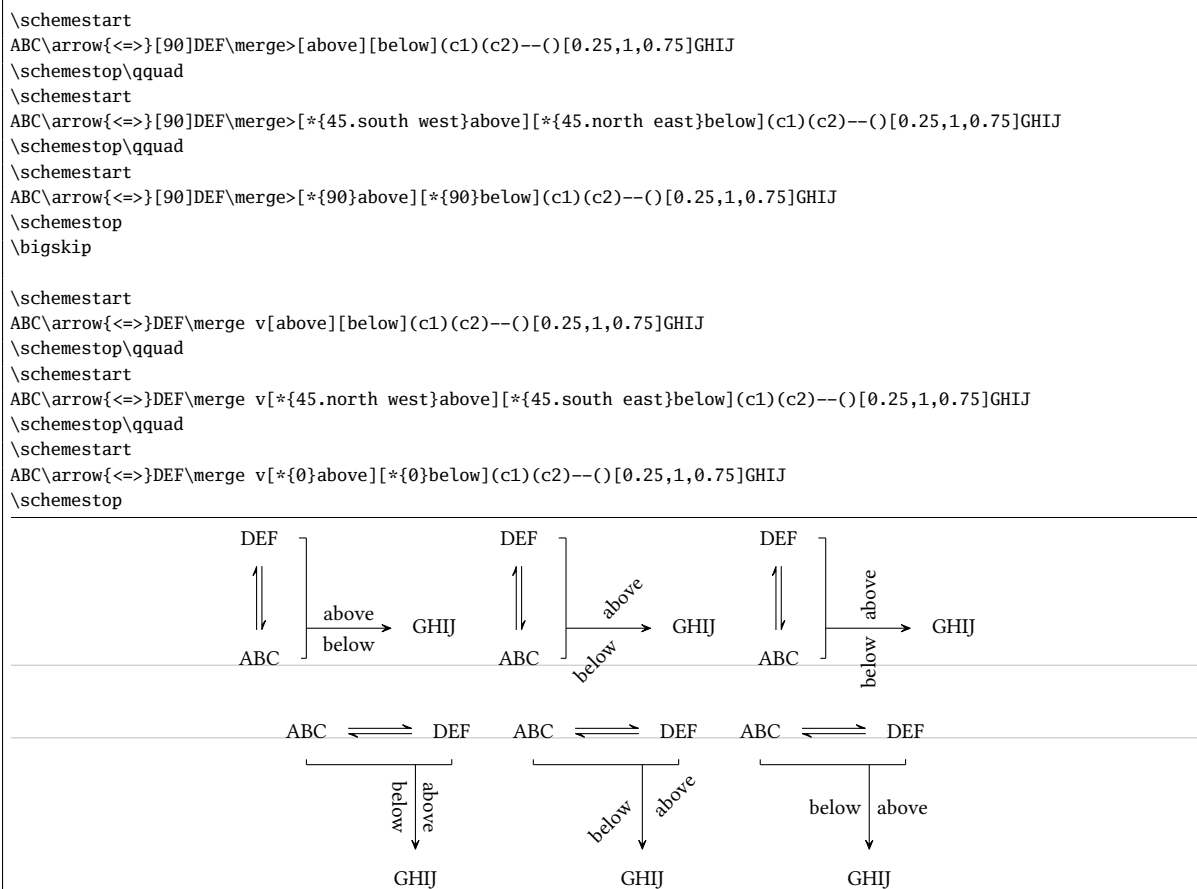


Finally, it is possible to write labels above or below the merged arrow. For this, the direction character accepts two optional arguments in brackets, a first one for the label above the arrow and a second one for the label below it. Therefore, the full syntax of the merge command is:

```
\merge{dir}[labelup][labeldown](n1.a1)(n2.a2)(...)(ni.ai)--(n.a[s])[c1,c2,c,style]
```

All the features introduced before for arrow labeling can be implemented here as well, i.e. rotation angle and anchoring with the syntax `*{angle.anchor}` entered just before the content of the label.

Labels of the \mergecommand

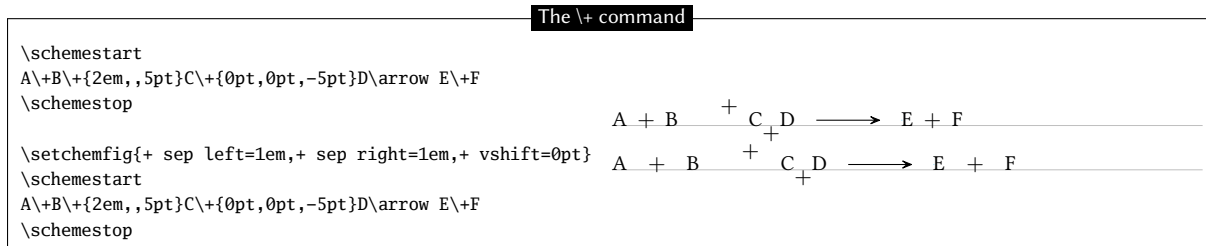


12 The + sign

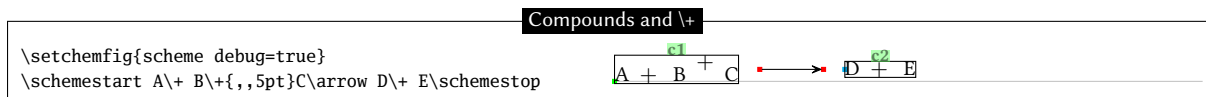
The use of a “\+” macro that displays a + sign is available between the commands `\schemestart` and `\schemestop`. This macro accepts an optional argument in braces with 3 dimensions in the form $\langle dim1 \rangle, \langle dim2 \rangle, \langle dim3 \rangle$, where:

- $\langle dim1 \rangle$ and $\langle dim2 \rangle$ are the dimensions to be inserted before and after the + sign;
- $\langle dim3 \rangle$ is the vertical offset of the sign.

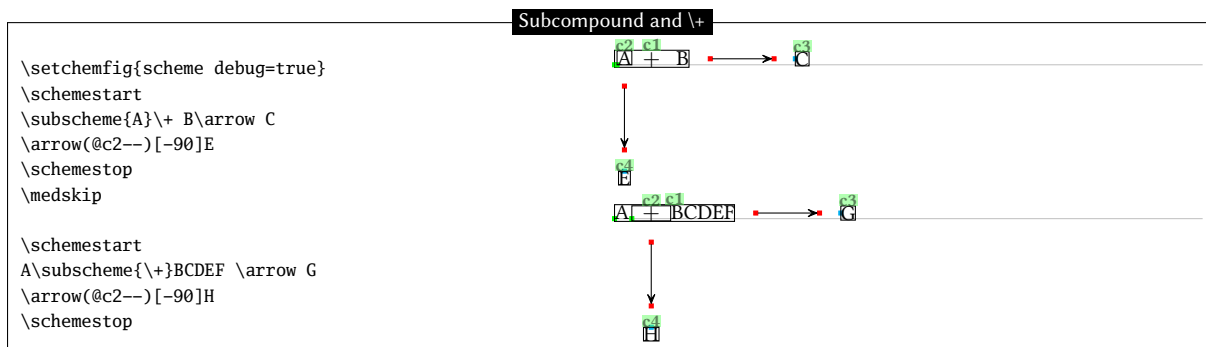
These dimensions can also be set, for all the following + signs with the $\langle keys \rangle$ `+ sep left = $\langle dim \rangle$` , `+ sep right = $\langle dim \rangle$` et `+ vshift = $\langle dim \rangle$` . The default values are 0.5em for the two first and 0pt for the third.



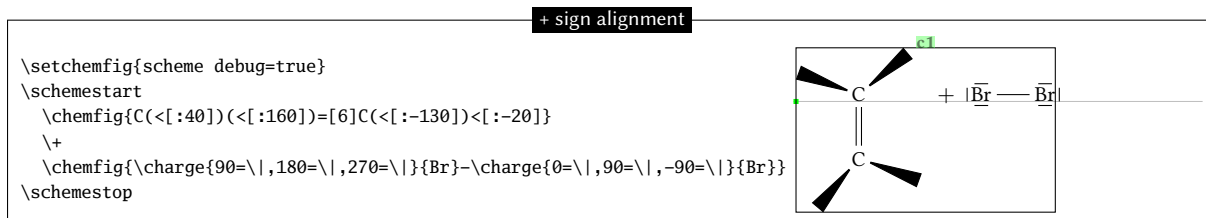
As shown in the example below, it should be kept in mind that the + sign inserted by the `\+` command is part of the compound:



This makes it difficult to draw a vertical arrow exactly below the letter “A” since this letter is not a single compound for `chemfig`. This issue can be solved with the use of the `\subscheme` command to uniquely define the letter “A” as a single compound (the same procedure can be applied to the + sign itself) so that it can be referred to later on with its own name:



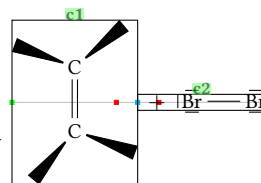
A common problem can be the misalignment of the “+” sign with the molecules before or after it. For example:



Here, the “+” sign sits on the same baseline as the compound before it, and this baseline is that of the top carbon atom. One may shift the “+” sign, but this would not change the vertical position of “|Br — Br|”. In fact, the “+” sign does not prevent `chemfig` from reading a compound, as shown in the example above where everything is included in the compound “c1”. Therefore, one must stop the compound right after the first molecule with a `\arrow{0}[,0]` that will draw an invisible, zero-length arrow. In order to vertically center the whole scheme, one must also set the the anchor of the first compound as “west” (or “180”, which is a synonym) with the second optional argument of the `\schemestart` command:

+ sign alignment

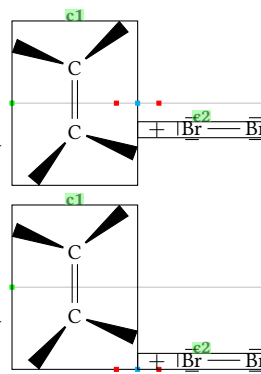
```
\setchemfig{scheme debug=true}
\schemestart[][west]
\chemfig{C(<[:40])<[:160])=[6]C<[:-130])<[:-20]}
\arrow{0}[,0]\+
\chemfig{\charge{90=\,180=\,270=\}{Br}-\charge{0=\,90=\,-90=\}{Br}}
\schemestop
```



Thus, the first compound “c1” consists of the first molecule and the second compound consists of everything else, i.e. the “+” sign and the second molecule. Alternatively, one can play with anchors or styles via the `\arrow` command to move the second compound to another location. Here, for example, the second compound is shifted downwards by 10pt in the first case. In the second case, the “south east” anchor of the first compound matches the “south west” anchor of the second one:

+ sign alignment

```
\setchemfig{scheme debug=true}
\schemestart[][west]
\chemfig{C(<[:40])<[:160])=[6]C<[:-130])<[:-20]}
\arrow(--[yshift=-10pt]){0}[,0]\+
\chemfig{\charge{90=\,180=\,270=\}{Br}-\charge{0=\,90=\,-90=\}{Br}}
\schemestop
\medskip
\schemestart[][west]
\chemfig{C(<[:40])<[:160])=[6]C<[:-130])<[:-20]}
\arrow(.south east--.south west){0}[,0]\+
\chemfig{\charge{90=\,180=\,270=\}{Br}-\charge{0=\,90=\,-90=\}{Br}}
\schemestop
```



List of commands

The commands created by chemfig are:

Commands	Description
<code>\chemfig[<i>settings</i>]{<i>code</i>}</code> <code>\setchemfig{<i>settings</i>}</code>	draws the molecule whose design is described by the <i>code</i> sets the parameters with the syntax <i>key</i> = <i>value</i> . Here is the full list of keys with the default values: <ul style="list-style-type: none"><code>chemfig style = \diamond</code>: style given to tikz<code>atom style = \diamond</code>: style of tikz nodes (atoms)<code>bond join = <i>false</i></code>: boolean for bond joins<code>fixed length = <i>false</i></code>: boolean for fixed bond widths<code>cram rectangle = <i>false</i></code>: boolean to draw rectangle Cram bond<code>cram width = <i>1.5ex</i></code>: length of the base of the triangles Cram bonds<code>cram dash width = <i>1pt</i></code>: width of dash Cram bonds<code>cram dash sep = <i>2pt</i></code>: space between dash Cram bonds<code>atom sep = <i>3em</i></code>: space between atoms<code>bond offset = <i>2pt</i></code>: space between atom and bond<code>double bond sep = <i>2pt</i></code>: space between multiple bonds lines<code>angle increment = <i>45</i></code>: increment of the angle of bonds<code>node style = \diamond</code>: style of atoms<code>bond style = \diamond</code>: style of bonds<code>cycle radius coeff = <i>0.75</i></code>: shrinkage ratio of the circle or arc inside cycles<code>stack sep = <i>1.5pt</i></code>: vertical gap between arguments of <code>\chemabove</code> and <code>\chembelow</code> macros<code>show cntcycle = <i>false</i></code>: show rings numbers<code>baseline = <i>0pt</i></code>: dimension or name of node to set the vertical position<code>debug = <i>false</i></code>: show atoms and groups of atoms<code>autoreset cntcycle = <i>true</i></code>: reset ring counter at <code>\chemfig</code> execution<code>compound style = \diamond</code>: style of compounds<code>compound sep = <i>5em</i></code>: space between compounds<code>arrow offset = <i>4pt</i></code>: space between compound and arrow<code>arrow angle = <i>0</i></code>: angle of the reaction arrow<code>arrow coeff = <i>1</i></code>: length ratio of arrows<code>arrow style = \diamond</code>: style of arrows<code>arrow double sep = <i>2pt</i></code>: space between double arrows<code>arrow double coeff = <i>0.6</i></code>: shrinkage ratio for the little arrow in "<code><-></code>" and "<code><-></code>"<code>arrow double harpoon = <i>true</i></code>: boolean for double harpoon arrows<code>arrow label sep = <i>3pt</i></code>: space between arrow and its label<code>arrow head = <i><CF</i></code>: style of arrow head<code>+ sep left = <i>0.5em</i></code>: space before the + sign<code>+ sep right = <i>0.5em</i></code>: space after the + sign<code>+ vshift = <i>0pt</i></code>: vertical shift of the + sign<code>gchemname = <i>true</i></code>: if true, makes the depth assignments made by <code>chemnameinit</code> and <code>chemname</code> global<code>schemestart code = \diamond</code>: code executed at the very beginning of a non-nested reaction scheme<code>schemestop code = \diamond</code>: code executed at the very end of a non-nested reaction scheme
<code>\resetchemfig</code>	Reset the parameters to their default values
<code>\printatom</code>	displays the atoms within the molecules. It can be redefined to customize the output. See page 26
<code>\hflipnext</code>	the next molecule will be horizontally flipped
<code>\vflipnext</code>	the next molecule will be vertically flipped
<code>\definesubmol{<i>name</i>}{<i>n</i>}[<i>code1</i>][<i>code2</i>]</code>	creates an alias <code>!<i>name</i></code> which can be put in the code of molecules to be drawn, and which will be replaced with <i>code1</i> or <i>code2</i> depending on the angle of the last bond. See page 28

<code>\chemskipalign</code>	tells the vertical alignment mechanism to ignore the current group of atoms. See page 33.
<code>\redefinesubmol{⟨name⟩}⟨n⟩[⟨code1⟩]{⟨code2⟩}</code>	replaces a preexisting alias !⟨name⟩ with the new ⟨code⟩. See page 29
<code>\charge[⟨parameters⟩]⟨pos⟩[⟨tikz⟩]{⟨atom⟩}</code>	prints ⟨atome⟩ and places the charges according to their ⟨positions⟩. The charges are placed out of the bounding box of the ⟨atome⟩. See page 33
<code>\Charge[⟨parameters⟩]⟨pos⟩[⟨tikz⟩]{⟨atom⟩}</code>	Same behaviour as <code>\charge</code> , but the final bounding box takes the charges into account.
<code>\chemmove[⟨tikz options⟩]⟨tikz code⟩</code>	Makes a <code>tikzpicture</code> 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 21.
<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 36
<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 36
<code>\Chemabove[⟨dim⟩]{⟨txt1⟩}{⟨txt2⟩}</code>	writes ⟨txt1⟩ and places ⟨txt2⟩ above, leaving ⟨dim⟩ of vertical space. See page 36
<code>\Chembelow[⟨dim⟩]{⟨txt1⟩}{⟨txt2⟩}</code>	writes {txt1} and places ⟨txt2⟩ below, leaving ⟨dim⟩ of vertical space. See page 36
<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.
<code>\schemestart... \schemestop</code>	commands between which a reaction scheme is drawn. See page 49.
<code>\arrow</code>	draws an arrow in a reaction scheme (this command is only defined inside a reaction scheme). See page 50.
<code>\+</code>	prints a + sign in a reaction scheme (this command is only defined inside a reaction scheme). See page 64.
<code>\subscheme{⟨code⟩}</code>	draws a subscheme (this command is only defined inside a reaction scheme). See 55.
<code>\definearrow</code>	defines an arrow. See page 59.
<code>\chemleft⟨car1⟩⟨stuff⟩\chemright⟨car2⟩</code>	draws expandable delimiters defined with ⟨car1⟩ and ⟨car2⟩ on the left and on the right of the ⟨stuff⟩, see page 55.
<code>\chemup⟨car1⟩⟨stuff⟩\chemdown⟨car2⟩</code>	draws expandable delimiters defined with ⟨car1⟩ and ⟨car2⟩ above and below the ⟨stuff⟩, see page 56.
<code>\polymerdelim[⟨settings⟩]{⟨node1⟩}{⟨node2⟩}</code>	draws delimiters at specified nodes, see page 46

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.

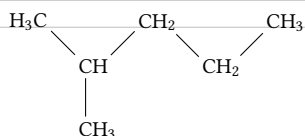
Anyway, I hope that this package will help all \LaTeX users wishing to draw molecules. Although `chemfig` has been thoroughly tested and although its version number is now greater than 1.0, I hope that you will be forgiving with bugs you encounter and send me an [email](#) to let me know of any malfunctions or suggestions for improvement.

Christian TELLECHEA

*
* *

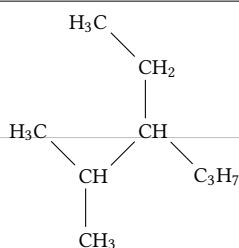
2-methylpentane

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



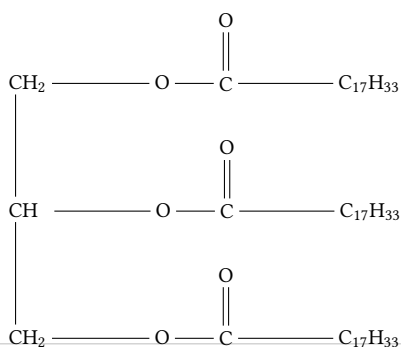
3-ethyl-2-methylhexane

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



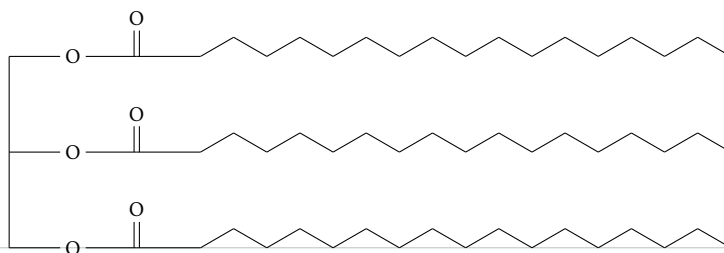
Stearine, condensed structural diagram

```
\definesubmol{@}{([0,2]-O-[0,1]C(=[2,1]O)-C_{17}H_{33})}  
\chemfig{[2,2]CH_2!@-CH_{\phantom 2}!@-CH_2!@}
```



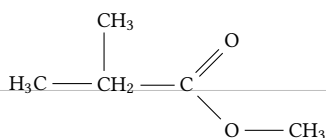
Stearine, skeleton diagram

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



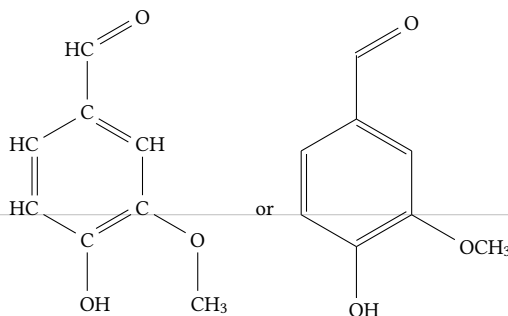
Methyl 2-methylpropanoate

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



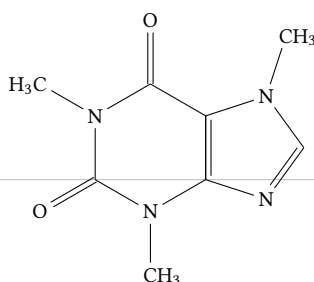
Vanillin

```
\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)-=)}
```



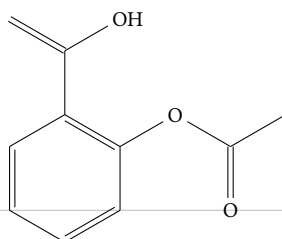
Caffeine

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



Aspirin

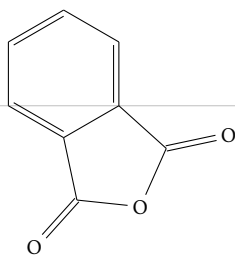
```
\chemfig{*6(=-(-O-[: -60])(=[: -60]O)-[: +60])=(-([ :+60])-[: -60]OH)-=)}
```



Aspirin is a registered trademark of Bayer in many countries.

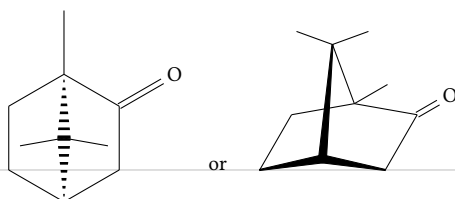
Phthalic anhydride

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



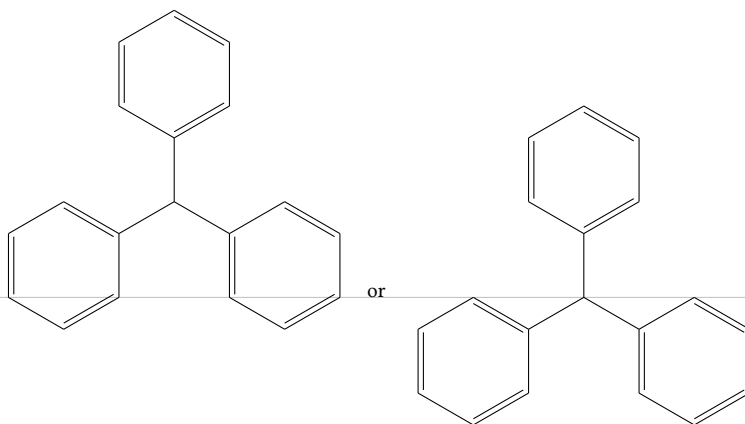
Camphor

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



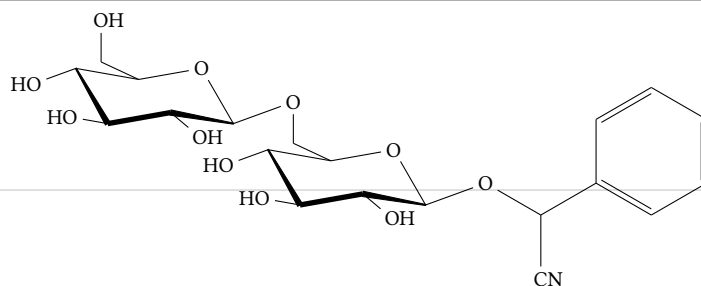
Triphenylmethane

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



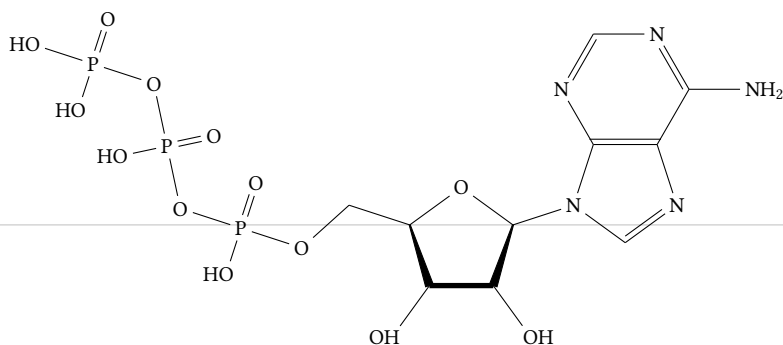
Amygdalin

```
\setchemfig{cram width=2pt}
\definesubmol{c1}{-[:200]-[:120]O-[: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{O(!{c1}(!{csub}O(!{c1}(!{csub}OH)!{c2}))!{c2})-[: -30](-[: -90]CN)-[:30]*6(-----)}
```



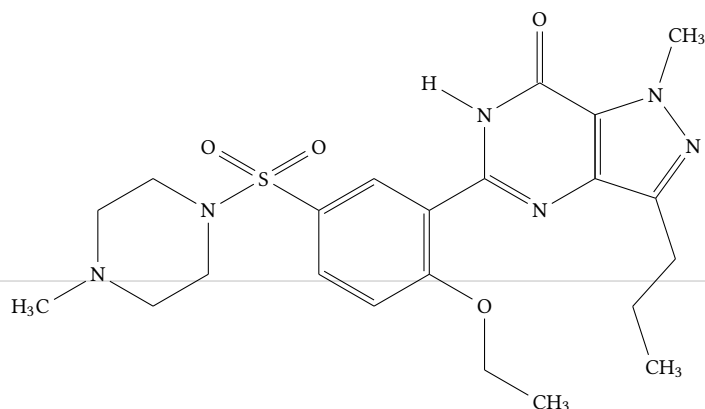
Adenosine triphosphate

```
\setchemfig{cram width=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-)}
```



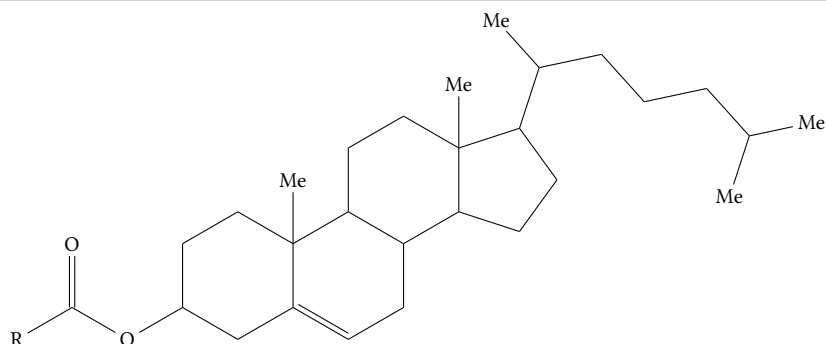
Viagra

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



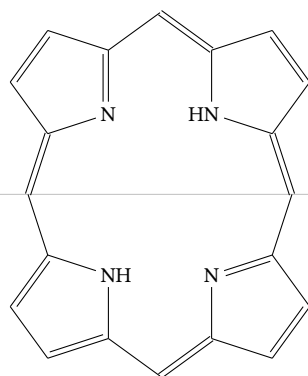
Cholesterol ester

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



Porphyrin

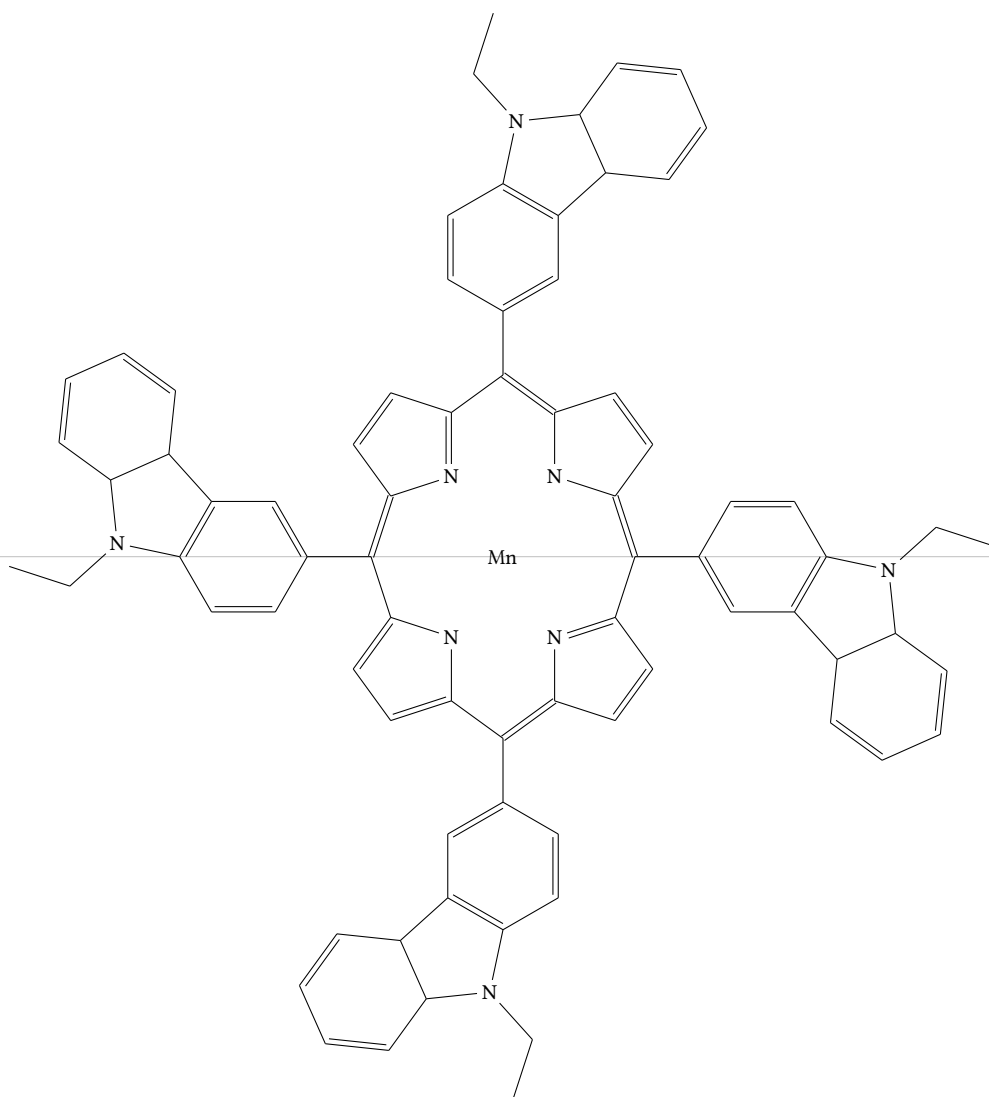
```
\chemfig{?=[::+72]*5(-N(-=[::-72]*5(-[,,,2]HN-[,,,2](-[::-36]*5(N(-=[::-72]*5(-NH-[,,,1]?=-=)
---))=-))=-)}
```



Manganese 5,10,15,20-tetra(N-ethyl-3-carbazolyl) porphyrin

```
\definesubmol{A}{*6(=*5(-*6(==--)--N(--[::-60])-)=--)}
\chemfig{([[:+180]-!A)=[[:+72]*5(-N(-[:+54]!A)=[[:72]*5(-N(-[::-33,1.5,,draw=none]Mn)
-(-[:+72]!A)-[:36]*5(=N(-[:+54]!A)-[:72]*5(-N(-)=))=--))=--)}

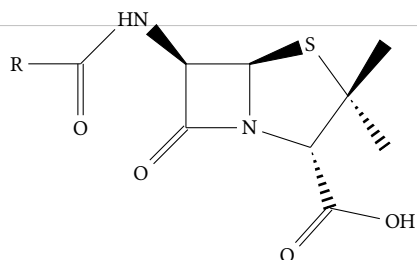
```



Penicillin

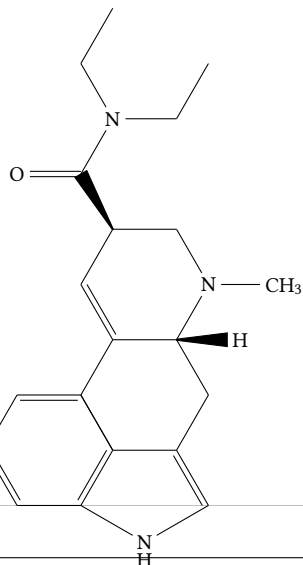
```
\chemfig{[:90]HN([::-45](-[:45]R)=[[:+45]O)>[:+45]*4(-=O)-N*5(-(<[:60]O)
-[:+60]OH)-(<[:+0](<[:108]-S>--))}

```

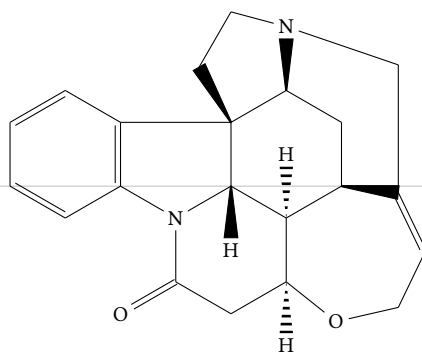
LSD

```
\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}?=))}
```



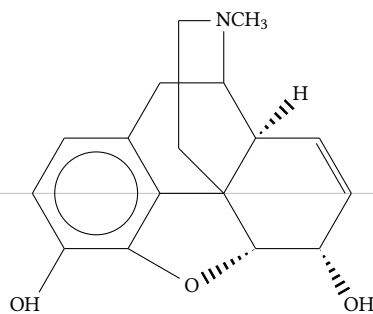
Strychnine

```
\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)--?)=?=-)}
```



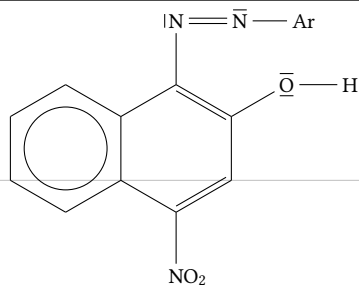
Codeine

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

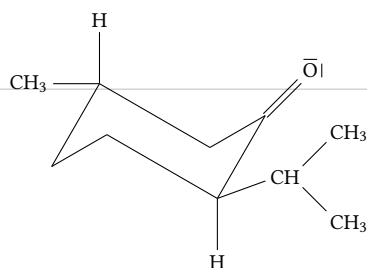


A dye (red)

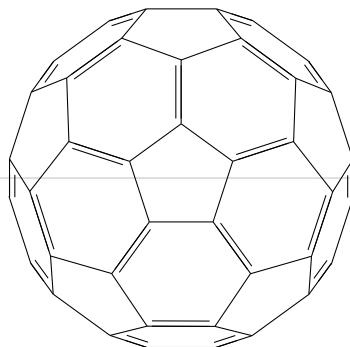
```
\chemfig{*6(--*6(-(-NO_2)-(-\charge{90=\|,-90=\|}{O})-[0]H)=(-\charge{180=\|}{N}=[0]\charge{90=\|}{N})-[0]Ar)-----)}
```

**Menthone**

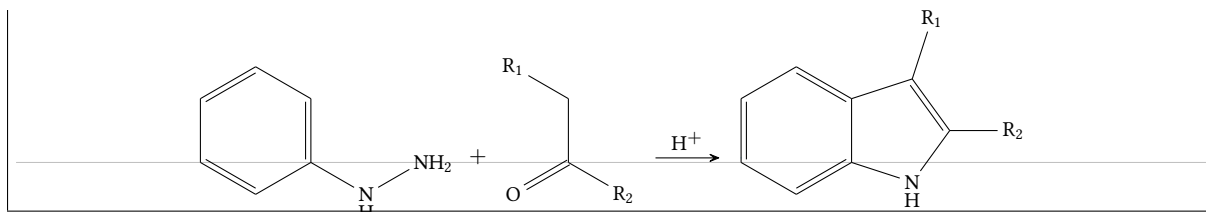
```
\chemfig{CH_3-?(-[2]H)(-[: -30,2]-[: +60](=[1]\charge{0=\|,90=\|}{O})-[: -150,1.5](-[:20]CH(-[1]CH_3)(-[7]CH_3))(-[6]H)-[: -90,2]-[: +60]?)}
```

**Fullerene**

```
\definesubmol\fragment1{
  (-[: #1,0.85,,draw=none]
  -[::126]-[::-54](=_(2pt,2pt)[::180])
  -[::-70](-[::-56.2,1.07]=^(2pt,2pt)[::180,1.07])
  -[::110,0.6](-[::-148,0.60](=^[::180,0.35])-[:::-18,1.1])
  -[::50,1.1](-[::18,0.60]=_[::180,0.35])
  -[::50,0.6]
  -[::110])
}
\chemfig{
  !\fragment{18}
  !\fragment{90}
  !\fragment{162}
  !\fragment{234}
  !\fragment{306}
}
```

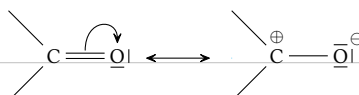
**Fischer indole synthesis**

```
\schemestart
  \chemfig{*6(--*6(-\chembelow{N}{H}-NH_2)---)}
  \+
  \chemfig{([:-150]O)(-[: -30]R_2)-[2]-[:150]R_1}
  \arrow(.mid east--.mid west){->[\chemfig{H^+}]}
  \chemfig{*6(--*5(-\chembelow{N}{H})-(-R_2)=(-R_1)----)}
\schemestop
```



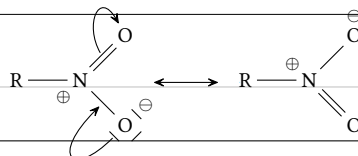
Reaction mechanisms: carbonyl group

```
\schemestart
\chemfig{C([3-])([5-])=[@{db,.5}]@{atoo}\charge{0=\,-90=\}|{0}}
\arrow(.mid east--.mid west){<->}
\chemfig{\charge{90:3pt=$\scriptstyle\oplus$}{C}([3-])([5-])-%
\charge{0=\,-90=\,-90=\,-45:3pt=$\scriptstyle\ominus$}{O}}
\schemestop
\chemmove{\draw[shorten <=2pt, shorten >=2pt](db) ..controls +(up:5mm) and +(up:5mm)..(atoo);}
```



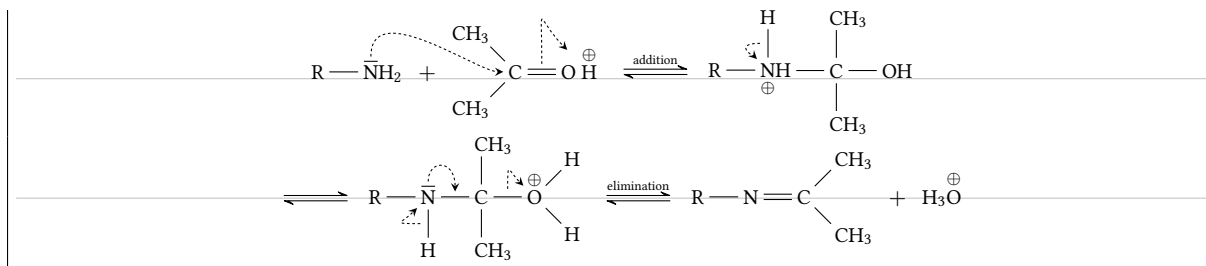
Reaction mechanisms: nitro group

```
\schemestart
\chemfig{R-\charge{225:3pt=$\scriptstyle\oplus$}{N}([1]=[@{db}]@{atoo1}O)([7]-[@{sb}]@{atoo2}
\charge{45=\,-45=\,-135=\,-45:5pt=$\scriptstyle\ominus$}{O})}
\arrow(.mid east--.mid west){<->}
\chemfig{R-\charge{135:3pt=$\scriptstyle\oplus$}{N}([1]-\charge{90:3pt=$\scriptstyle\ominus$}{O})([7]=O)}
\schemestop
\chemmove{
\draw[shorten <=2pt, shorten >=2pt](db) ..controls +(120:5mm) and +(120:7mm)..(atoo1);
\draw[shorten <=3pt, shorten >=2pt](atoo2) ..controls +(225:10mm) and +(225:10mm)..(sb);
}
```



Nucleophilic addition. Primary amines

```
\setchemfig{atom sep=2.5em,compound sep=5em}
\schemestart
\chemfig{R-@{aton}\charge{90=\}|{N}H_2}
\+
\chemfig{@{atoc}C([3]-CH_3)([5]-CH_3)=[@{atoo1}]O}
\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);}
\arrow{<=>[\tiny addition]}
\chemfig{R-@{aton}\chembelow{N}{\scriptstyle\oplus}H([2]-[@{sb}]H)-C(-[2]CH_3)-([6]CH_3)-OH}
\schemestop
\chemmove{
\draw[-stealth,dash pattern= on 1pt off 1pt,shorten <=3pt, shorten >=2pt]
(sb)..controls +(left:5mm) and +(135:2mm)..(aton);}
\par
\schemestart
\arrow{<=>}
\chemfig{R-@{aton}\charge{90=\}|{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);}
\arrow{<=>[\tiny elimination]}\chemfig{R-N=C(-[1]CH_3)(-[7]CH_3)}
\+
\chemfig{H_3\chemabove{O}{\scriptstyle\oplus}}
\schemestop
```

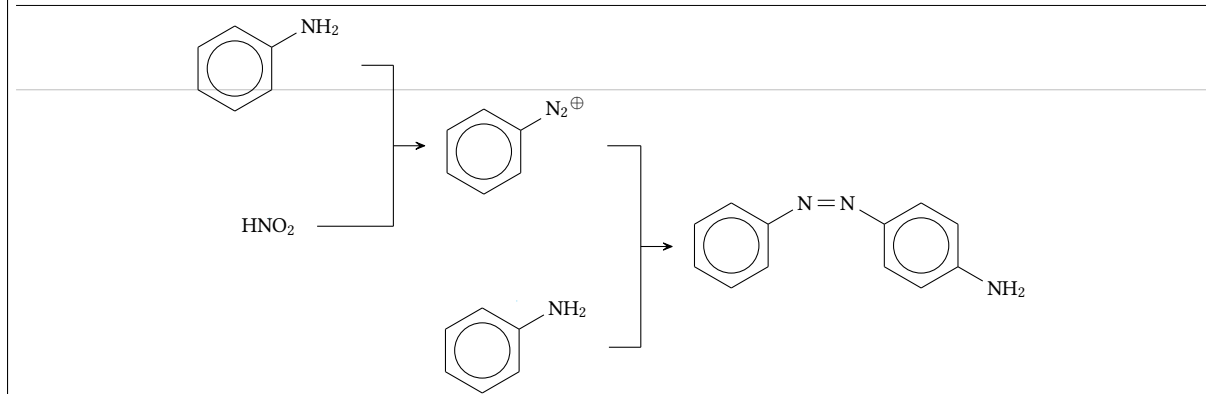


Reaction scheme

```

\setchemfig{atom sep=2em}
\schemestart[-90]
  \chemfig{*6(---(-NH_2)---)}\arrow{0}\chemfig{HNO_2}
  \merge(c1)(c2)--()
  \chemfig{*6(---(-N_2|{}^\oplus)---)}\arrow{0}\chemfig{*6(---(-NH_2)---)}
  \merge(c3)(c4)--()
  \chemfig{*6(---(-N=[::-30]N-[::-30])*6(---(-NH_2)---))---)}
\schemestop

```

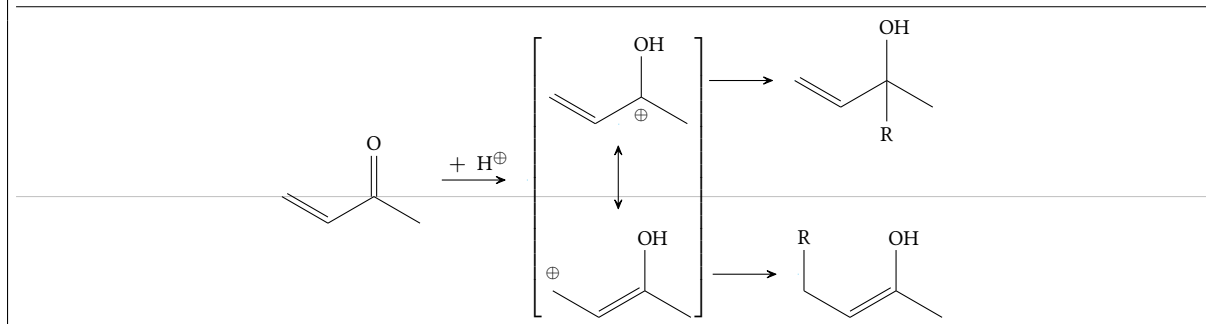


Addition

```

\setchemfig{atom sep=2.5em}
\schemestart
  \chemfig{*6(=-(=([2]O))}
  \arrow{->[\+\chemfig{H^\oplus}]}
  \chemleft[\\subscheme[90]{%
    \chemfig{*6(=[2,0.33,,draw=none]\scriptstyle\oplus)-(-)-OH)}
  \arrow{<->}
  \chemfig{*6(=-(=([6,0.33,,draw=none]\scriptstyle\oplus)-OH))}\chemright}
  \arrow{@c3--}\chemfig{*6(=[2]R)-(-)-OH)}
  \arrow{@c4--}\chemfig{*6(=-(=([6]R)-OH)}
\schemestop

```



Electrophilic aromatic substitution

```

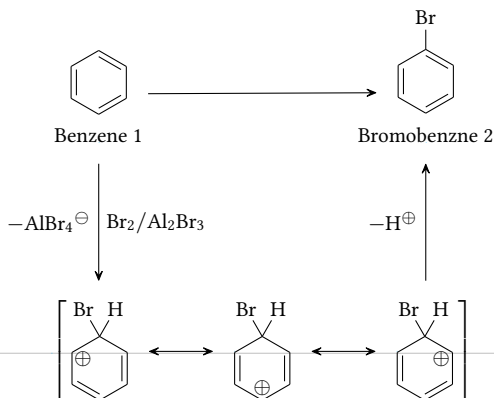
\setchemfig{atom sep=1.5em}%
\definesubmol{+}{-[-0.4,,draw=none]\oplus}%
\schemestart
  \arrow{0}{,0}
  \chemleft[\\subscheme{\chemfig{*6(=---(-[:120]Br)-[:60]H)-(!+)-)}
  \arrow{<->}
  \chemfig{*6(-(!+)----(-[:120]Br)-[:60]H)-=)}
  \arrow{<->}

```

```

\chemfig{*6(==(!+)-(-[:120]Br)(-[:60]H)-)}\chemright]
\arrow{@c2--}{<-[*0\chemfig{-}AlBr_4|^{\ominus}][*0\chemfig{Br_2/Al_2Br_3}][90,1.5]}
\chemname{\chemfig{*6(==---)}\{Benzene 1}
\arrow{@c4--}{->[*0\chemfig{-}H^{\oplus}][90,1.5]}
\chemname{\chemfig{*6(==(-Br)-)}\{Bromobenzne 2}
\arrow{@c5.mid east--@c6.mid west}
\schemestop
\chemnameinit{}

```

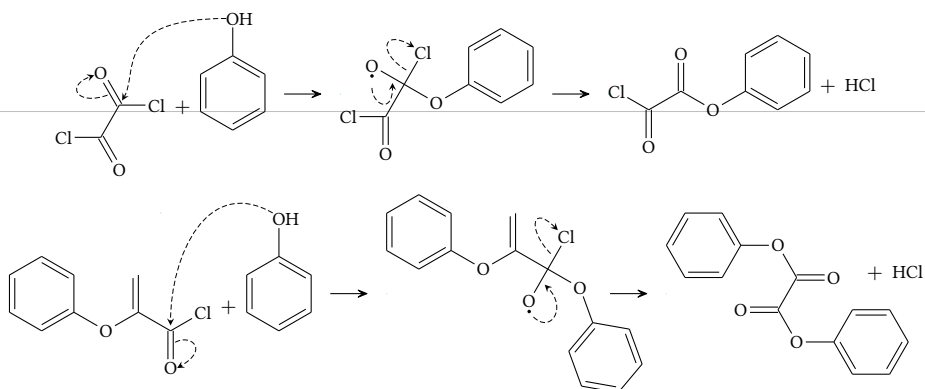


Reaction mechanism of chlorination

```

\scriptsize\setchemfig{bond offset=1pt,atom sep=2em,compound sep=4em}
\schemestart
\chemfig{Cl-[4]@{a0}(=[@{a1}:120]@{a2}O)-[:120](=[:-60]O)-[4]Cl}\+\chemfig{*6(==(-@{oh1}OH)-)}\arrow
\chemfig{*6((-[:150](-@{o0}:150)@{o1}\charge{-90=\.}{O})(-[:60]@{c1}Cl)-[:240](-[4]Cl)=[6]O)===)}
\arrow\chemfig{*6((-[:150]([2]O)-[:150]([6]O)-[:150]Cl)===)}\+\chemfig{HCl}
\arrow{@c1--}{0}[-90,0,5]
\chemfig{*6(==6(-@{o2}([@{o3}]@{o4}O)-Cl)===)}\+\chemfig{*6(==(-@{oh2}OH)-)}\arrow
\chemfig{*6(==6(-(-[:150]@{c1}Cl)-[:60]@{c2}Cl)-[:120]@{o5}\charge{-90=\.}{O})-[:40]*6(==---))=)}
\kern-3em \arrow\chemfig{[:30]*6(=-(-[:60]([=O]-[:120]([4]O)-[:60]O)-*6(==---))=)}
\kern-3em \+\chemfig{HCl}
\schemestop
\chemmove[linewidth=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)..(c11.150);
\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

```

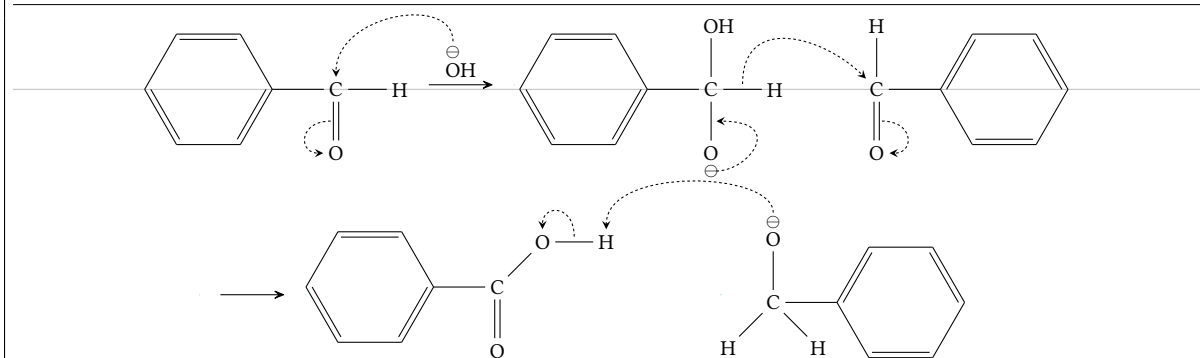
\schemestart
\chemfig{[:30]*6(==(-@{atoc}C([6]=[@{db}]@{atoo1}O)-H)-)}
\arrow(start.mid east--.mid west){->[\chemfig{@{atoo2}\chemabove{O}{\scriptstyle\ominus}}H]}
\chemmove[-stealth,shorten >=2pt,dash pattern=on 1pt off 1pt,thin]{
\draw[shorten <=8pt](atoo2)..controls+(up:10mm)and+(up:10mm)..(atoc);
\draw[shorten <=2pt](db)..controls+(left:5mm)and+(west:5mm)..(atoo1);}
\chemfig{[:30]*6(==(-C([6]-[@{sb1}]@{atoo1}\chembelow{O}{\scriptstyle\ominus})([2]-OH)-[@{sb2}]H)-)}
\hspace{1cm}

```

```

\chemfig{[:30]*6((-@{atoc}C[6]=[@{db}]@{atoo2}O)-[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);}
\arrow{@start.base west--}{0}[-75,2]
}
\arrow
\chemfig{[:30]*6(==(-C[1]-@{atoo2}O-@{sb}0@{atoh}H)([6]=O))-)}
\arrow{0}
\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);}
\schemestop

```

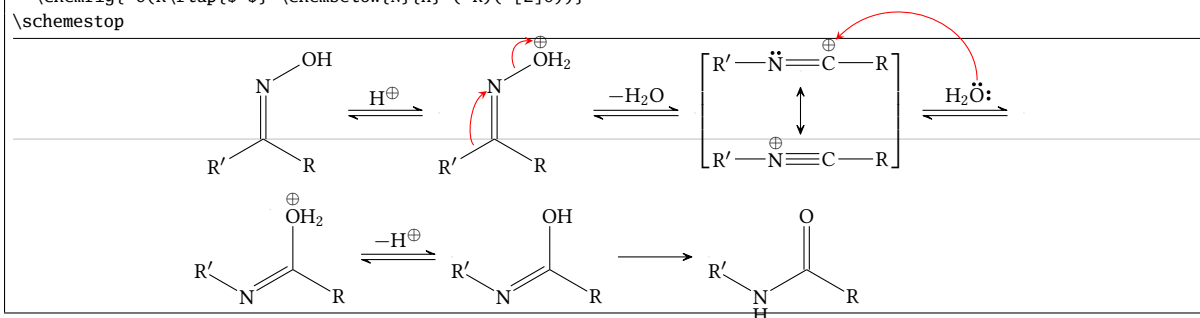


Beckmann rearrangement

```

\setchemfig{bond offset=1pt,atom sep=2.5em,compound sep=5em,arrow offset=6pt}
\schemestart
\chemfig{(-[:150]R')(-[:30]R)=[2]N-[:30]OH}
\arrow{<=>[\chemfig{H^oplus}]}
\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);}
\arrow{<=>[\chemfig{{-}H_2O}]}[1,1]
\chemleft[\subscheme[90]{%
  \chemfig{R'-\chemabove{N}{\scriptstyle\oplus}~C-R}
  \arrow{<->}[0.75]
  \chemfig{R'-\charge{90=:\}{N}=@{a1}\chemabove{C}{\scriptstyle\oplus}-R}\chemright]
\arrow{<=>[\chemfig{H_2O@{a0}\charge{0=:\,90=:\}{O}}]}[1,1]
\chemmove[red,-stealth,red,shorten <=3pt]{
  \draw(a0)..controls+(90:10mm)and+(45:10mm)..([yshift=6pt]a1.45);}
\arrow{@c1--}{0}[-90,0.333]
\chemfig{*6(R\rlap{ '$' }-N(-R)-\chemabove{0}{\scriptstyle\oplus} H_2)}
\arrow{<=>[\chemfig{{-}H^oplus}]}
\chemfig{*6(R\rlap{ '$' }-N(-R)-OH)}
\arrow
\chemfig{*6(R\rlap{ '$' }-\chembelow{N}{H}-(-R)(=[:2]O))}
\schemestop

```



Reaction scheme

```

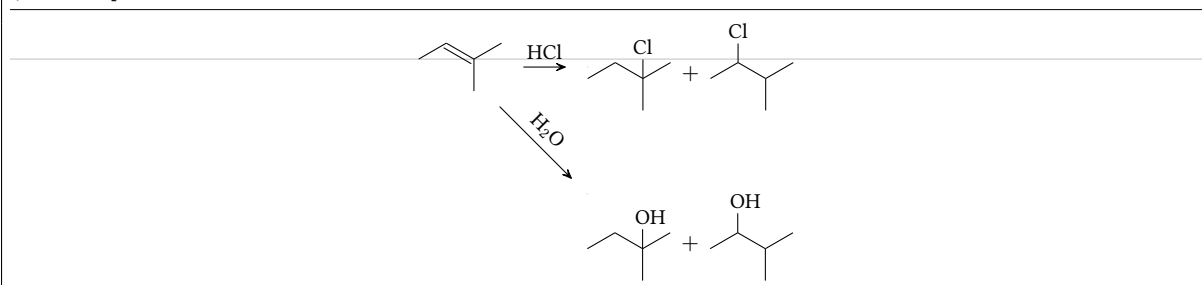
\setchemfig{atom sep=1.5em,compound sep=4em}
\schemestart
\chemfig{-[:30]=[:60](-[:60])-[:60]}
\arrow{->[\chemfig{HCl}]}
\chemfig{-[:30]-[:60](-[:120]Cl)(-[:60])-[:60]}\chemfig{-[:30](-[:60]Cl)-[:60](-[:60])-[:60]}

```

```

\arrow@c1--.north west}{->[\chemfig{H_2O}]}[-45,1.7]
\chemfig{-[:30]-[:60](-[:120]OH)(-[:60]-[:60])+\chemfig{-[:30](-[:60]OH)-[:60](-[:60]-[:60])}
\schemestop

```

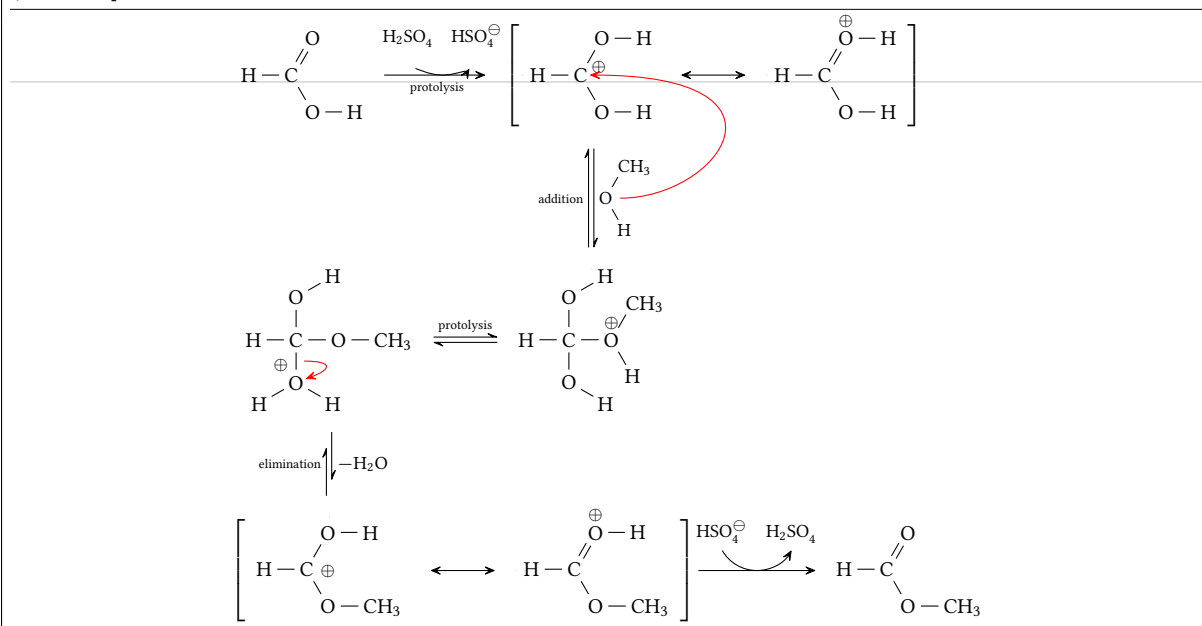


Esterification of formic acid

```

\tikzset{obrace/.style={left delimiter={[]},inner sep=3pt},
cbrace/.style={right delimiter={}],inner sep=3pt},
braces/.style={left delimiter={[]},right delimiter={}],inner sep=3pt}}
\setchemfig{atom sep=2em}
\schemestart
\chemfig{H-C(=[:60]O)-[:60]O-H}
\arrow{--M1[obrace]}{-U>[\scriptsize\chemfig{H_2SO_4^+}][\scriptsize\chemfig{HSO_4^ominus}][.25]}%
[.15,shorten >=6pt]
\chemfig{H-@{a2}C(-[:60]O-H)(-[:30].5,,draw=none){\scriptstyle\oplus}-[:60]O-H}
\arrow{--[cbrace]}{<->}
\chemfig{H-C(=[:60]\chemabove{O}{\scriptstyle\oplus})-H)-[:60]O-H}
\arrow{@M1--}{<=>[\scriptsize\chemfig{H-[:120]@{a1}O-[:60]CH_3}][*\{0\}\tiny addition]}[-90,1.33]
\chemfig{H-C(-[:2]O-[:30]H)(-\chemabove{O}{\scriptstyle\oplus})(-[:60]CH_3)-[:60]O-[:30]H}
\arrow{<=>[\tiny protolysis]}[180]
\chemfig{H-C(-[:2]O-[:30]H)(-O-CH_3)-@{b1}6@{a3}\chemabove{O}{\kern-4mm\scriptstyle\oplus})(-[:150]H)-[:30]H}
\arrow{--[obrace]}{<=>[\scriptsize\chemfig{-[:H_2O]}][*\{0\}\tiny elimination]}[-90,,shorten >=6pt]
\chemfig{H-C(-[:60]O-H)(-[:.5,,draw=none){\scriptstyle\oplus})-[:60]O-CH_3}
\arrow{--[cbrace]}{<->}
\chemfig{H-C(=[:60]\chemabove{O}{\scriptstyle\oplus})-H)-[:60]O-CH_3}
\arrow{-U>[\scriptsize\chemfig{HSO_4^ominus}][\scriptsize\chemfig{H_2SO_4^+}][.25]}[.15,5]
\chemfig{H-C(=[:60]O)-[:60]O-CH_3}
\arrow{@M1--[yshift=-5pt]}{0}[180,.5]{\tiny protolysis}
\chemmove[red,shorten <=3pt,shorten >=1pt]{
\draw(a1)..controls +(0:1.5cm)and+(0:3cm).. (a2);
\draw(b1)..controls +(0:5mm)and+(20:5mm).. (a3);}
\schemestop

```



Electrophilic addition of halogen to olefin

```

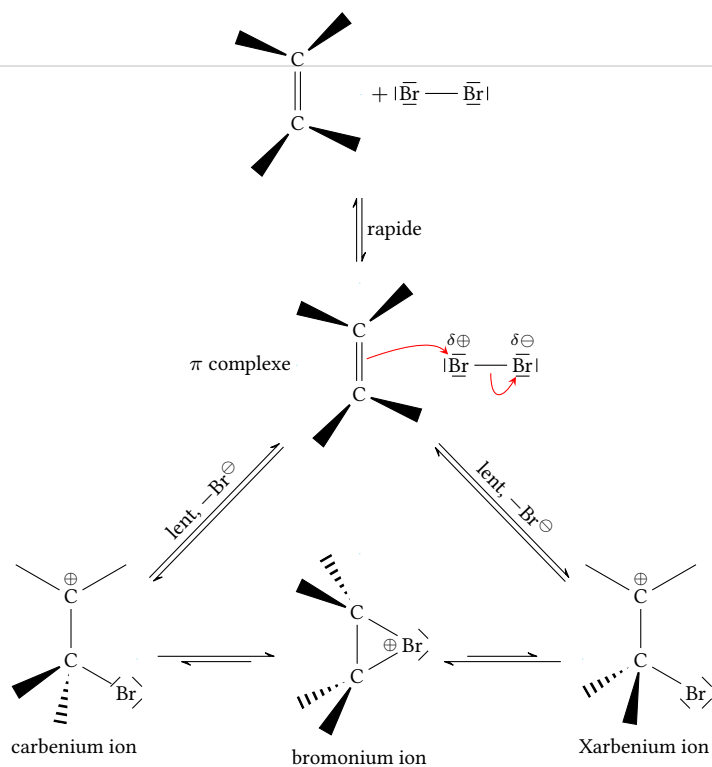
\schemestart
\subscheme%
\chemfig{C(<[:40])(<[:160])=[6]C(<[:130])<[:20]}
\arrow{0}{,0}+\chemfig{\charge{90=\,180=\,270=\}{Br}-\charge{90=\,0=\,270=\}{Br}}

```

```

\arrow{@c1--olefin}{<=>[*{0}rapide]}[-90]
\chemfig{>[: -20]C<[: 40]}=[@{db}6]C<[: -130]<[: -20]}
\arrow{--bromonium}{0}[-90]
\chemname{\chemfig{C*3((<[: -155])-\charge{45=\|,-45=\|,180:3pt=$\scriptstyle\oplus$\{Br}-C(<[: 155])-)}\{bromonium ion}}
\arrow{--carbeniumA}{<=>}[1.5]
\chemname{\chemfig{-[: -30]\chemabove{C}{\scriptstyle\oplus}(-[: 30])-[6]C<[: -150])<[: -100])-[: -30]\charge{45=\|,-45=\|,225=\|}\{Br}\}\{Xarbenium ion}}
\arrow{@bromonium--carbeniumB}{<=>}[180,1.5]
\chemname{\chemfig{-[: -30]\chemabove{C}{\scriptstyle\oplus}(-[: 30])-[6]C<[: -150])<[: -100])-[: -30]\charge{45=\|,-45=\|,135=\|}\{Br}\}\{carbenium ion}}
\arrow{@olefin--}{0}[.25]
\chemfig{@{Br1}\charge{90=\|,180=\|,270=\|,90:5pt=$\scriptstyle\delta\oplus$\{Br}-[@{b2}]@{Br2}\charge{90=\|,0=\|,270=\|,90:5pt=$\scriptstyle\delta\ominus$\{Br}}}}
\arrow{@olefin--[left]}{0}[180,0]
$\pi$ complexe
\arrow{@carbeniumA--@olefin}{<=>[lent, \chemfig{{-}Br^\ominus}]}
\arrow{@carbeniumB--@olefin}{<=>[lent, \chemfig{{-}Br^\ominus}]}
\chemmove[-stealth,red,shorten <=3pt,shorten >=2pt]{
\draw(db) .. controls +(20:5mm) and +(135:5mm) .. (Br1);
\draw(b2) .. controls +(-90:5mm) and +(-120:5mm) .. (Br2);}
\schemestop
\chemnameinit{}

```

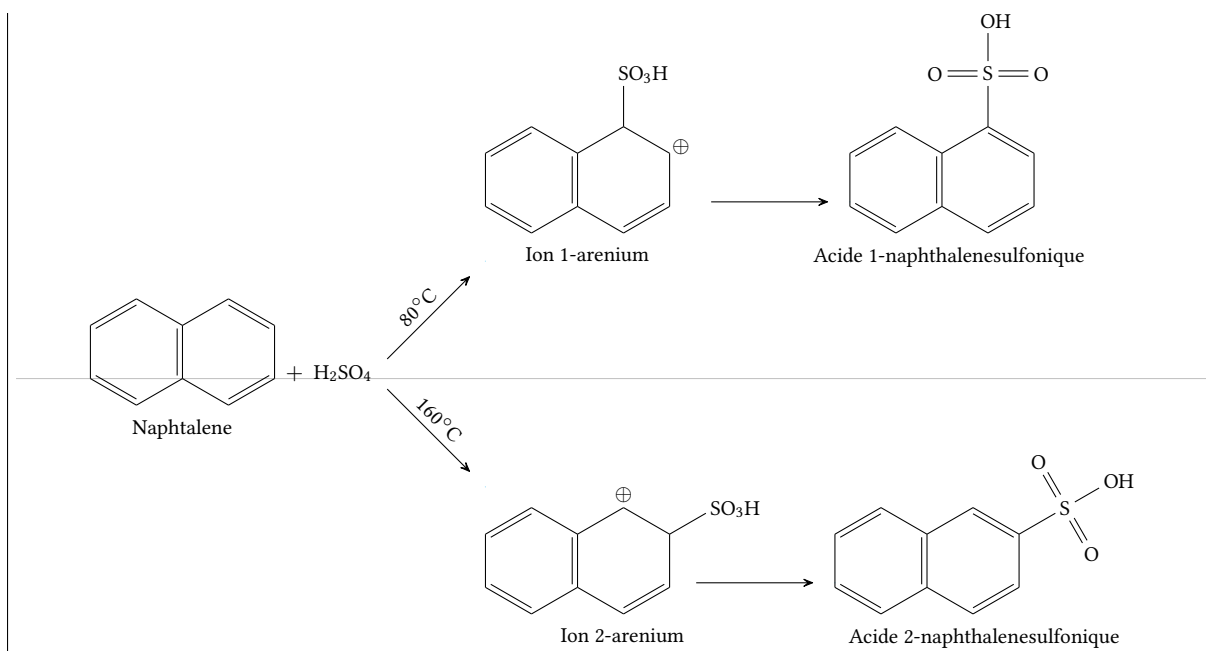


Sulfonation of naphthalene

```

\definesubmol\cycloplus{-[,0.25,,draw=none]\oplus}
\definesubmol{so2oh}{S(=[: 90]O)(=[: -90]O)-OH}
\setchemfig{atom sep=2.5em}
\schemestart[,1.5]
\chemname{\chemfig{*6(=*6(==*)=*)}\{Naphtalene}\+\chemfig{H_2SO_4}}
\arrow{nph.mid east--.south west}{->[80\degrees C]}[45]
\chemname{\chemfig{*6(=*6(-(!\cycloplus)-(-SO_3H)-)=*)}\{Ion 1-arenium}}
\arrow{.mid east--.mid west}
\chemname{\chemfig{*6(=*6(-=(!so2oh)-)=*)}\{Acide 1-naphthalenesulfonique}}
\arrow{@nph.mid east--.north west}{->[160\degrees C]}[-45]
\chemname{\chemfig{*6(=*6(-(-SO_3H)-(!\cycloplus)-)=*)}\{Ion 2-arenium}\kern-4em}
\arrow{.mid east--.mid west}
\chemname{\chemfig{*6(=*6(-(-!so2oh)-)=*)}\{Acide 2-naphthalenesulfonique}}
\schemestop
\chemnameinit{}

```

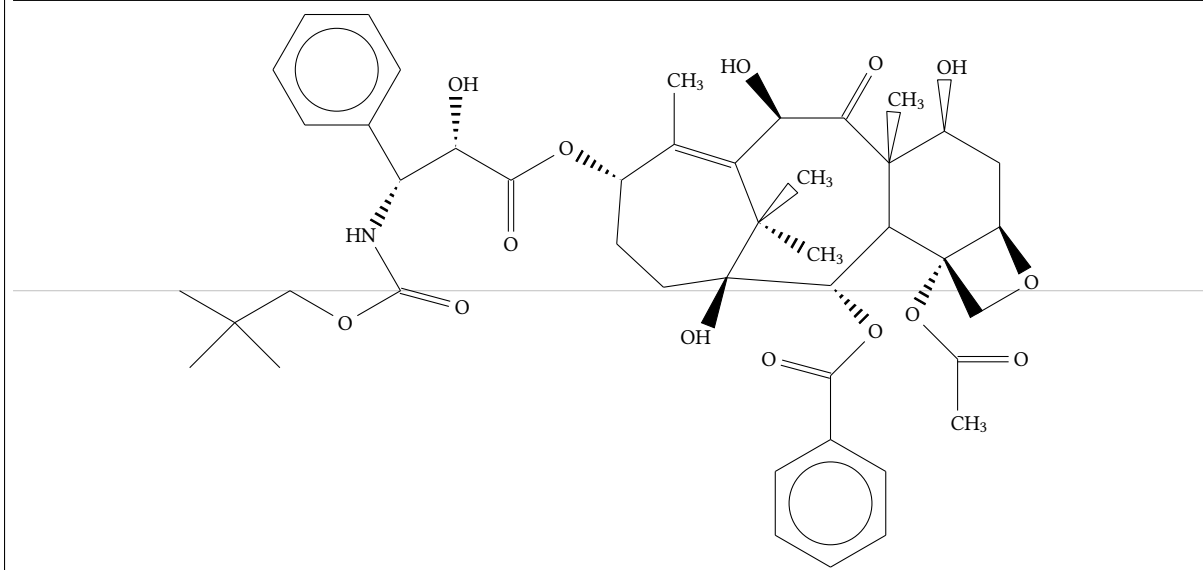



Taxotere

```

\chemfig{(-[:30](-[5])(-[7])-[:+60]-[:60]O-[:+60](=[:45]O)-[:+90]HN>[:60](-[:+60]**6(-----))
-[:30](<:[2]OH)-[:60](=[6]O)-[:+60]O>[:60]**7(---?(<[:120]OH)-(<[1]CH_3)<[:90]CH_3)
-([1](<[:+80]HO)-[0](=[:+60]O)-[7](<[:+130]CH_3)-[:+75](<[2]OH)-[:60]-[:60](<[:+30]O-[:90])
-[:60](<[:+90])(<[:+30]O-[7](-[6]CH_3)=[0]O)-[:60]-[6]-[5,1,3]?(<[7]O-[5](=[:60]O)
-[6]**6(-----))=(-[2]CH_3-)}

```

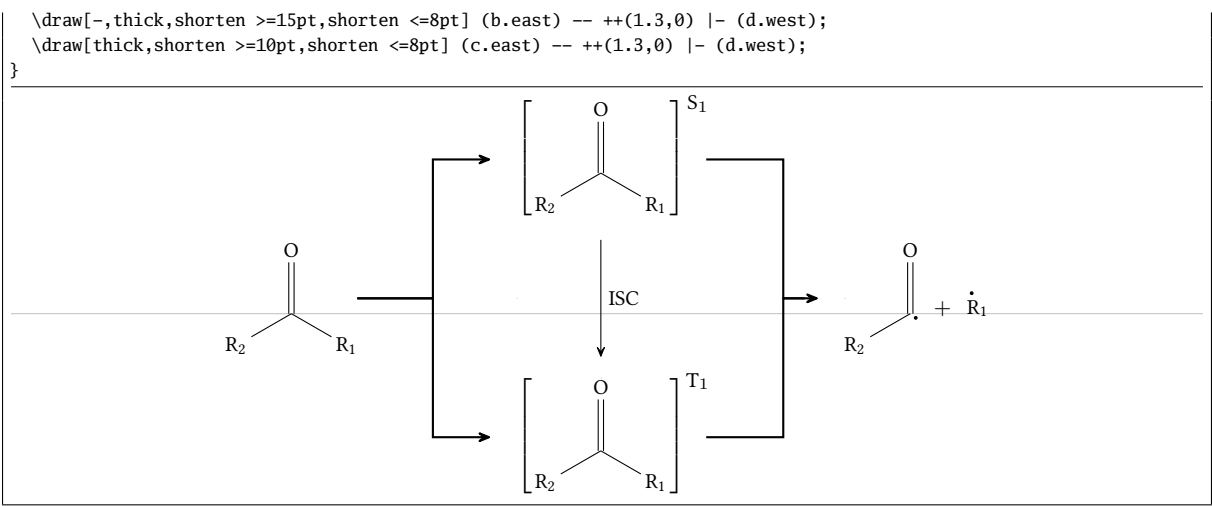


Diverging arrows

```

\schemestart
\chemfig{(-[:210]R_2)(-[:330]R_1)=[2]O}
\arrow(a--)[,1.5,,,draw=none]
\subscheme{
\charge{30:4pt}=\mathrm{S}_1$\chemleft{\chemfig{(-[:210]R_2)(-[:330]R_1)=[2]O}\chemright{}}
\arrow(b--c){->[*0]ISC}}[-90,1.5]
\charge{30:4pt}=\mathrm{T}_1$\chemleft{\chemfig{(-[:210]R_2)(-[:330]R_1)=[2]O}\chemright{}}
}
\arrow(--d)[,1.5,,,draw=none]
\chemfig{(-[:210]R_2)(-[:330,0.1,,,draw=none]\charge{330:-1pt}=\.,)}=[2]O}
\+
\chemfig{\charge{90:1pt}=\.,}{R}_1}
\schemestop
\chemmove{
\draw[thick,shorten >=10pt] (a.east) -- ++(1,0) |- (b.west);
\draw[thick,shorten >=10pt] (a.east) -- ++(1,0) |- (c.west);
}

```



ANEXO C – MORBI ULTRICES RUTRUM LOREM.

Sed mattis, erat sit amet gravida malesuada, elit augue egestas diam, tempus scelerisque nunc nisl vitae libero. Sed consequat feugiat massa. Nunc porta, eros in eleifend varius, erat leo rutrum dui, non convallis lectus orci ut nibh. Sed lorem massa, nonummy quis, egestas id, condimentum at, nisl. Maecenas at nibh. Aliquam et augue at nunc pellentesque ullamcorper. Duis nisl nibh, laoreet suscipit, convallis ut, rutrum id, enim. Phasellus odio. Nulla nulla elit, molestie non, scelerisque at, vestibulum eu, nulla. Ut odio nisl, facilisis id, mollis et, scelerisque nec, enim. Aenean sem leo, pellentesque sit amet, scelerisque sit amet, vehicula pellentesque, sapien.

**ANEXO D – CRAS NON URNA SED FEUGIAT CUM SOCIIS NATOQUE
PENATIBUS ET MAGNIS DIS PARTURIENT MONTES NASCETUR RIDICULUS MUS**

Sed consequat tellus et tortor. Ut tempor laoreet quam. Nullam id wisi a libero tristique semper. Nullam nisl massa, rutrum ut, egestas semper, mollis id, leo. Nulla ac massa eu risus blandit mattis. Mauris ut nunc. In hac habitasse platea dictumst. Aliquam eget tortor. Quisque dapibus pede in erat. Nunc enim. In dui nulla, commodo at, consectetuer nec, malesuada nec, elit. Aliquam ornare tellus eu urna. Sed nec metus. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas.

ANEXO E – FUSCE FACILISIS LACINIA DUI

Phasellus id magna. Duis malesuada interdum arcu. Integer metus. Morbi pulvinar pellentesque mi. Suspendisse sed est eu magna molestie egestas. Quisque mi lorem, pulvinar eget, egestas quis, luctus at, ante. Proin auctor vehicula purus. Fusce ac nisl aliquam ante hendrerit pellentesque. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Morbi wisi. Etiam arcu mauris, facilisis sed, eleifend non, nonummy ut, pede. Cras ut lacus tempor metus mollis placerat. Vivamus eu tortor vel metus interdum malesuada.