



Pour démarrer avec L^AT_EX

Version du 12 juin 2017

Sommaire

1	Installation et premiers conseils	1
I	Principes généraux	1
II	Les distributions	1
III	Éditeurs de textes	1
IV	Afficher le fichier .pdf obtenu après compilation dans TexMaker	2
V	Personnalisation de TexMaker	3
VI	Installer un nouveau package non présent dans la distribution	4
A	« À la main »	4
B	Vérifier si un package est présent dans la distribution avec TeX Live Manager	5
VII	Sites sur lesquels on trouve des documents écrits avec \LaTeX	6
VIII	Logiciels ou sites permettant d'extraire du code \LaTeX	7
2	Premier document	8
I	Mon premier document \LaTeX	8
A	Écrire un fichier source	8
B	Premières remarques	9
II	Explication rapide du code source	9
A	Structure du code source	9
B	Explication du préambule	9
III	Commandes	10
A	Arguments d'une commande	10
B	Commandes semi-globales	11
C	Les packages	12
IV	Les caractères spéciaux	12
3	Mise en forme	13
I	Mise en forme de base	13
A	Un premier exemple	13
B	Police et fontes	13
C	Changement de la taille des fontes	14
D	Alignement	15
E	Espaces	16
II	Couleur	18

III	Mise en page	19
A	Dimensions de la page	19
B	Multicolonnes	19
IV	Structurer un document	21
A	Listes structurées	21
B	Sectionnement	22
C	Références croisées	23
V	Renvois	24
4	Mathématiques	25
I	Un premier exemple	25
A	Utilisation de <code>\usepackage[autolanguage,np]{numprint}</code>	26
B	Utilisation de <code>\usepackage{xlop}</code>	26
C	Utilisation de <code>\usepackage{cancel}</code>	27
D	Utilisation <code>\usepackage{dsfont}</code>	27
E	Fontes mathématiques	28
II	Les modes mathématiques	28
A	En ligne ou hors du texte ?	28
B	Du texte dans des formules	29
C	Les espaces	30
III	Écrire des mathématiques de la sixième à la terminale	31
A	Au collège	31
B	Au lycée	35
C	En licence	38
IV	Aligner des égalités	38
V	Écrire un système	39
5	Tableaux	40
I	Composer un tableau	40
A	Créer des lignes et des colonnes	40
B	Fusionner des colonnes	41
C	Spécificateurs de colonne supplémentaires	41
II	Les tableaux en mathématiques	42
A	L'environnement <code>array</code>	42
B	Les systèmes	42
C	Les matrices	43
III	Tableaux de signes et tableaux de variations	44
A	Tableaux de signes	44
B	Tableaux de variations	45
C	Un mélange	46
D	Un dernier exemple	46
IV	Complément : le tableur	47

6	Images	49
I	Insérer une image	49
II	Modifier une image	49
III	Les environnements <i>wrapfigure</i> et <i>minipage</i>	50
IV	Une image dans du texte	51
7	Graphiques et géométrie	53
I	Différentes méthodes	53
II	Présentation générale de TikZ et TkZ	53
III	Exemples de figures utilisant TikZ	54
IV	Exemples en ligne	60
V	Exemples de figures utilisant TkZ	60
8	Arbres pondérés	73
9	QR codes	78

Installation et premiers conseils

I Principes généraux

Pour utiliser \LaTeX , il faut installer, dans l'ordre,

- ★ une distribution
- ★ un éditeur de textes

On tape un document appelé *code source* dans l'éditeur de textes et on l'enregistre au format `.tex`. On *compile* ce document pour obtenir un document lisible, en général au format `.pdf`. La compilation utilise la distribution \LaTeX que vous avez installée.

Lors de l'installation d'une distribution, je vous conseille d'effectuer une installation complète de la distribution, c'est-à-dire avec tous les *packages*.

Mais si malgré cela, il vous manque des *packages*, vous les trouverez directement sur le site de l'auteur du *package* et sur la « banque des *packages* \LaTeX » : <https://www.ctan.org/>.

En général, l'auteur du *package* fournit une notice de ce *package* et on retrouve cette notice sur le site <https://www.ctan.org/>.

Si vous souhaitez ajouter un *package* à votre distribution, reportez vous à la dernière section de ce chapitre.

II Les distributions

Il existe des distributions \LaTeX pour *Linux*, *Windows* ou *Mac*.

La distribution *TeXLive* existe pour *Linux* et *Windows*.

La distribution *MiKTeX* est une distribution uniquement pour *Windows*.

La distribution *MacTeX* est une distribution uniquement pour *Mac*.

Personnellement, j'utilise la distribution *TeXLive* sur *Linux* et *Windows*.

L'installation de ces distributions est très bien détaillée sur le site

<http://www.xmlmath.net/doculatem/index.html>.

Pour ceux qui souhaitent installer *TeXLive* mais n'ont pas de DVD sous la main, comme indiqué sur le site précédent, il existe d'autres possibilités :

- ★ installation à partir du site <http://tug.org/texlive/acquire-netinstall.html> (en Anglais)
- ★ utilisation du logiciel *Daemon Tools* qui permet, sous *Windows*, à partir d'une image `.iso` de créer un DVD virtuel
- ★ ou montage, sous *Linux* de l'image `.iso`

III Éditeurs de textes

On peut utiliser n'importe quel éditeur de textes pour utiliser \LaTeX . Il suffit d'enregistrer le *fichier source* avec comme extension `.tex`.

Mais, je vous conseille d'utiliser un éditeur de textes spécifique à \LaTeX .

Il en existe plusieurs, dont, par exemple *TeXmaker* ou *TeXStudio*.

Personnellement, j'utilise le premier des deux qui fonctionne sous toutes les plateformes (*Linux*, *Windows* ou *Mac*).

Un éditeur spécifique à \LaTeX vous permettra, entre autres...

- ★ d'avoir le *code source* colorié automatiquement en fonction des commandes que vous utiliser,
- ★ d'utiliser l'auto-complétion pour les commandes usuelles de \LaTeX ,
- ★ d'ajouter d'autres commandes (y compris celles que vous avez créées) dans le « vocabulaire » du logiciel et ainsi utiliser l'auto-complétion pour ces commandes,
- ★ de créer des raccourcis clavier pour des bouts de code que vous utilisez souvent,
- ★ de paramétrer de différentes façons la compilation de votre document.

L'installation de *TeXmaker* est détaillée sur le site précédent :

<http://www.xmlmath.net/doculatem/index.html>

Remarque :

Il existe également des éditeurs en ligne (dans un navigateur) qui ont l'avantage de pouvoir compiler le document au fur et à mesure qu'on le tape.

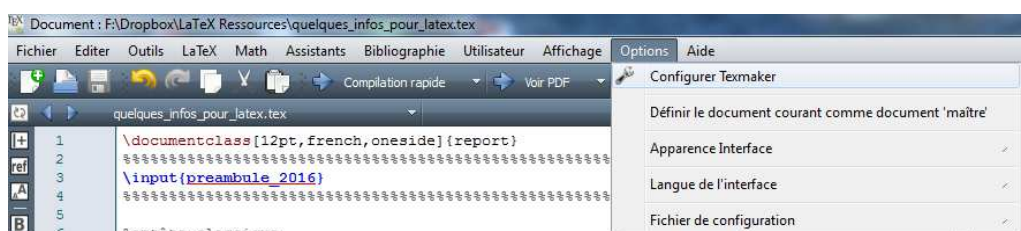
On peut utiliser par exemple <https://www.overleaf.com/> qui est un très bon éditeur en ligne.

Il suffira alors de créer un compte gratuitement pour pouvoir créer des documents avec un espace de 100 Mb au départ.

IV Afficher le fichier .pdf obtenu après compilation dans TexMaker

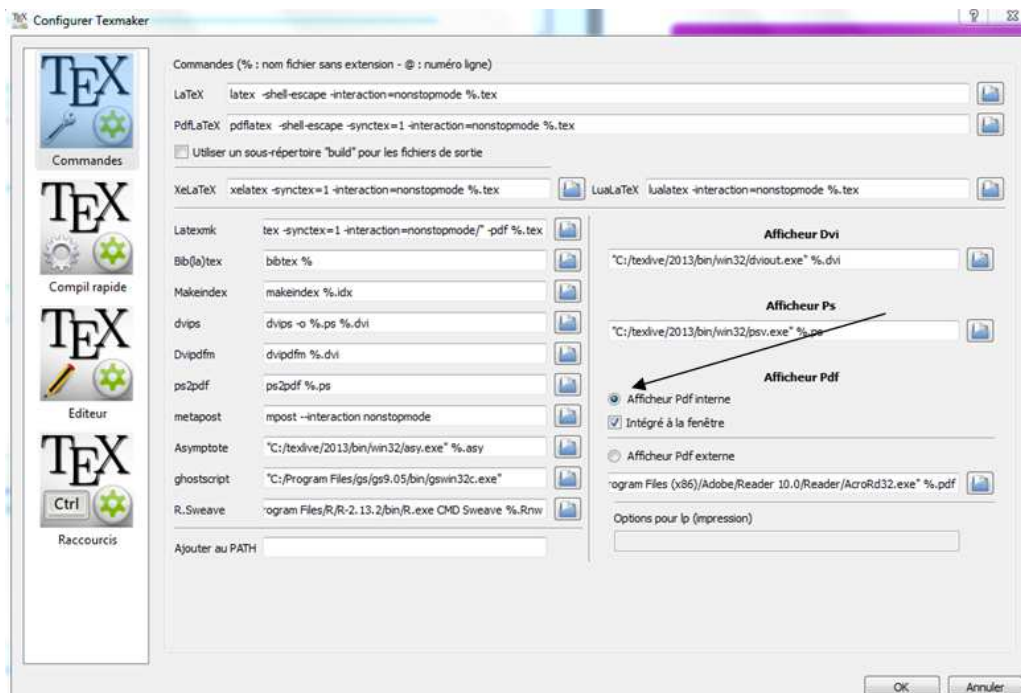
On suit les étapes suivantes :

- ★ Dans le menu « Options », on clique sur « Configurer Texmaker » :



- ★ On affiche l'écran suivant en choisissant « Commandes » dans la barre gauche de la fenêtre qui vient de s'ouvrir.

On clique ensuite sur « Afficher Pdf interne » et on coche « Intégré à la fenêtre » :



On valide en cliquant sur OK.

★ Il suffit, pour terminer, de cocher sur « Pdf Viewer » en bas à gauche :



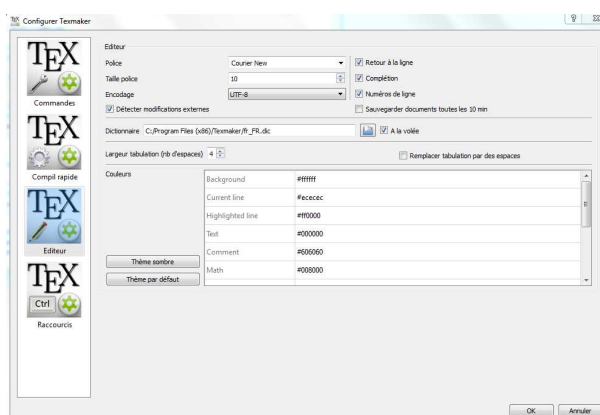
V Personnalisation de TexMaker

★ Dans le menu « Utilisateur »,

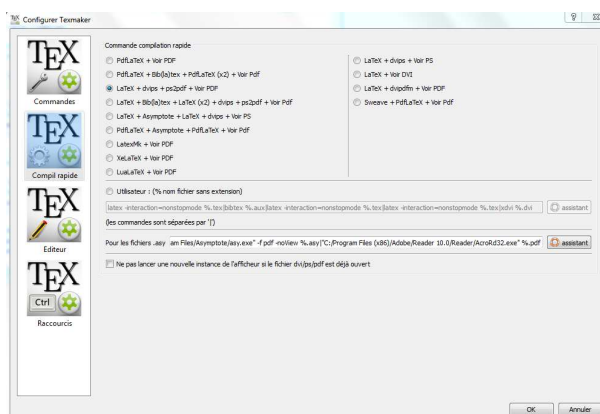
- on peut créer des balises,
- ajouter des commandes,
- ajouter des termes au « dictionnaire » initial reconnu par le logiciel qu'il utilise pour effectuer de l'auto-complétion.

★ Dans le menu « Options », en cliquant sur « Configurer Texmaker »,

- on peut choisir le système d'encodage de l'éditeur



- ainsi que la méthode de compilation (accessible par la touche **F1**)



Si le document final doit contenir des figures PSTricks et TikZ, préférez la méthode de compilation de la capture d'écran précédente.

VI Installer un nouveau package non présent dans la distribution

A « À la main »

Il existe plusieurs méthodes d'installation d'un package avec LaTeX. J'ai sélectionné pour vous les deux plus faciles à mon sens. Elles devraient vous permettre d'utiliser la quasi-totalité des packages.

Les deux méthodes développées ici diffèrent légèrement, suivant que votre package est un fichier .ins ou .sty.

Dans de rares cas, les packages sont fournis sous d'autres extensions, mais ils sont alors accompagnés d'un fichier README vous guidant lors de leur installation.

1 Les packages en .sty, méthode simple

Si votre *package* est de la forme `nom_de_package.sty`, rien de plus simple pour l'utiliser : il suffit de le copier dans le dossier contenant votre source `.tex`. Lorsque votre distribution compilera le fichier `.tex`, elle recherchera dans ce dossier les fichiers `.sty` des packages manquants, et le tour sera joué.

Résumons, la commande `\usepackage{nom_de_package}` demande à LaTeX d'utiliser un *package* installé ou, s'il ne l'est pas, d'aller chercher le fichier `nom_de_package.sty` dans le dossier de travail.

Simple, n'est-ce pas ?

Mais si l'on veut pouvoir réutiliser ce même *package* pour d'autres documents sans avoir à la copier dans le dossier de travail, on le copiera dans le dossier du PC contenant tous les *packages* :

1. Sur mon installation, il s'agit du répertoire :

- a) `C:\texlive\2013\texmf-dist\tex\latex` pour mon pc sous Windows 7
- b) `/usr/local/texlive/2013/texmf-dist/tex/latex` pour mon netbook sous Linux

2. Il faut ensuite demander à la distribution installée de prendre en compte cet ajout :

a) Sous Windows

Lancer « Tex Live Manager », puis dans le menu « Actions », cliquer sur « Update filename database »

OU

Lancer l'invite de commandes et taper « texhash » (sans les guillemets...), puis valider.

b) Sous Linux

Lancer un terminal en administrateur et taper « texhash » (sans les guillemets), puis valider.

2 Les packages en .ins, méthode en deux temps

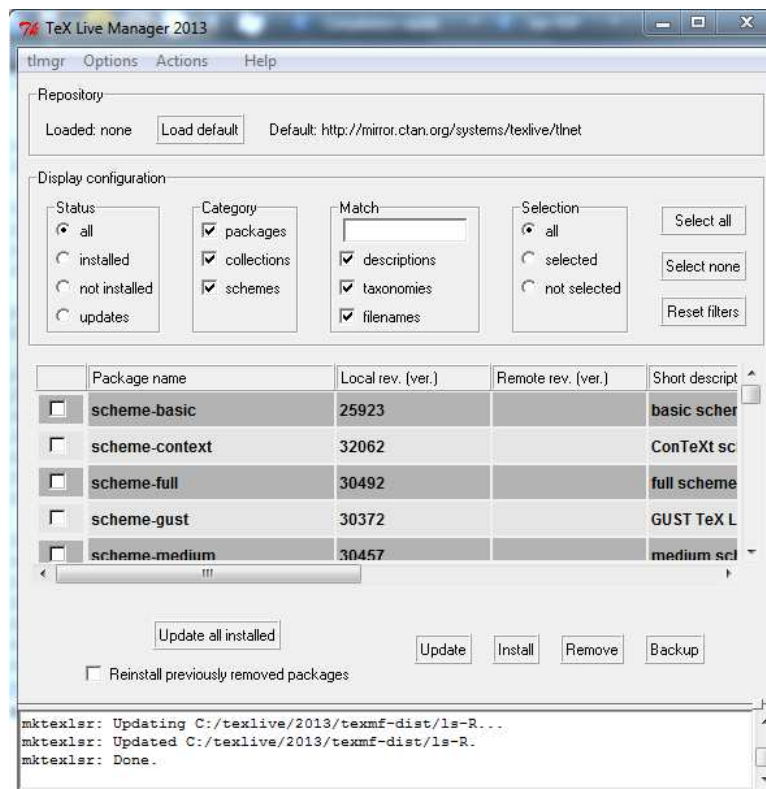
Les *packages* contenus dans un fichier `.ins` doivent être traités en deux étapes.

Premièrement, mettez votre fichier `nom_de_package.ins` dans un répertoire et compilez-le (c'est-à-dire, lancer la commande `latex nom_de_package.ins` en ligne de commandes) : il générera un fichier `nom_de_package.sty`.

Ce fichier `nom_de_package.sty` doit être traité selon le processus développé dans le paragraphe « Les packages en .sty, méthode simple ».

B Vérifier si un package est présent dans la distribution avec TeX Live Manager

On lance Tex Live Manager et on voit alors apparaître la liste de tous les *packages* présents dans la distribution.



On peut également utiliser la ligne « Match » en tapant les premières lettres du nom de *package* cherché pour accélérer la recherche.

VII Sites sur lesquels on trouve des documents écrits avec \LaTeX

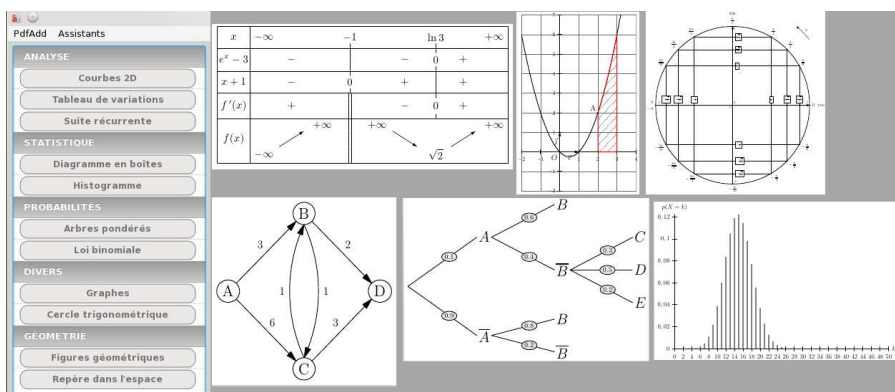
★ Parmi les sites sur lesquels on trouve des exercices de Mathématiques écrits avec \LaTeX , on peut citer [Sesamath](#) dont les manuels pour le lycée utilisent \LaTeX , ou l'incontournable site de l'[APMEP](#) avec sa section de sujets d'annales pour le brevet, le bac, le bts etc.

★ Et parmi les très nombreux sites sur lesquels on trouve de la documentation sur l'utilisation de \LaTeX orientée « Mathématiques », on peut citer :

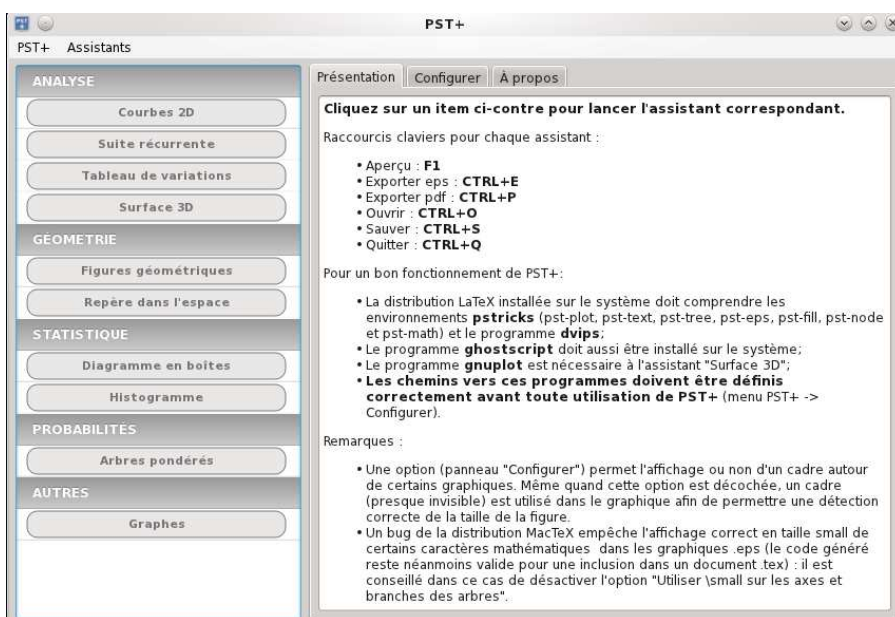
- la [section dédiée](#) sur le site de l'Irem de l'académie de Lyon qui contient une brochure intitulée « LaTeX... pour le prof de maths! » ,
- le site [xmlmath](#) de Pascal Brachet (déjà cité plus tôt),
- le site [math.et.info](#) avec sa rubrique sur TikZ (très utile pour les figures et graphiques (voir le chapitre 7)),
- le site [altermundus](#) qui contient de l'aide pour l'utilisation de TikZ et TkZ pour les graphiques et figures (voir le chapitre 7),
- et le site [texample.net](#) sur lequel on retrouve énormément de figures créées avec TikZ.

VIII Logiciels ou sites permettant d'extraire du code \LaTeX

★ Certains logiciels permettent d'exporter dans un format utilisable par \LaTeX les graphiques créés. C'est le cas d'**Algobox**, de **Geogebra**, de **PdfAdd**, qui permet de créer les graphiques ci-dessous, en générant du code Asymptote (ou \LaTeX pour les tableaux de variations/signes) qu'il suffit de copier/coller dans son document \LaTeX :



C'est également le cas de **Pst+**, qui permet de créer les graphiques ci-dessous, en générant du code \LaTeX /PSTricks qu'il suffit de copier/coller dans son document \LaTeX :



★ On peut également intégrer du code \LaTeX dans la plupart des blogs, ou directement dans une zone graphique du logiciel **Geogebra**.

★ Sur le site math.et.info, on peut créer des tableaux de variations et des arbres pondérés, puis récupérer le code TikZ à copier/coller dans le document \LaTeX .

Premier document

I Mon premier document L^AT_EX

A Écrire un fichier source

Dans la pratique, ce que l'on écrit avec *Texmaker* se nomme le *fichier source* (ou encore le *code source*). Ce fichier est compilé par L^AT_EX (plusieurs fois si nécessaire) et les différentes lignes de code sont interprétées pour obtenir en bout de chaîne le document final, celui qui sera imprimé.

Voici un premier document que vous pouvez taper dans votre éditeur, enregistrer, puis compiler (touche **F1** sur *TeXmaker*).

```
_____ Premier document _____
1 \documentclass[12pt,french]{article}
2 \usepackage[utf8x]{inputenc}
3 \usepackage[T1]{fontenc}
4 \usepackage[upright]{fourier}
5 \usepackage[a4paper,margin=2cm]{geometry}
6 \usepackage{mathtools,amssymb}
7 \usepackage{babel}
8
9 \begin{document}
10 \textbf{Définition 1.} Pour tous réels  $a$ ,  $b$  et  $c \neq 0$ , on a : \par
11  $\frac{a}{c} + \frac{b}{c} = \frac{a+b}{c}$ . % Facile !
12
13 \textbf{Définition 2.} Soit  $x \geq 0$  et  $A \geq 0$  :
14  $\sqrt{x} = A \Leftrightarrow A^2 = x$ . % Evident !
15
16
17  $\sqrt[3]{x} = A \Leftrightarrow A^3 = x$ . % marche aussi avec  $x < 0$  !
18
19 Évidemment,
20 ce
21 n'est
22 pas bien compliqué.
23 \end{document}
24 C'est vraiment trop bien !!
_____
```



En règle générale, dans les noms des documents, on évitera d'utiliser des espaces et des lettres accentuées.

B Premières remarques

1. Le symbole `$` sert à ouvrir et fermer le mode mathématiques (qui permet d'écrire des formules mathématiques).
2. Le symbole `%` permet d'écrire des commentaires qui n'apparaîtront pas dans le document compilé.
3. La commande `\par` de la ligne 10 sert à effectuer un saut de ligne dans le document compilé. Une ligne vide a le même effet.
Personnellement, je préfère la deuxième solution qui me permet d'aérer mon *code source*.
4. Un simple changement de ligne dans le *code source* ne sert qu'à aérer le *code source* et n'a aucun effet sur le document final.
5. Dans le *code source*, lorsque plusieurs espaces séparent deux mots, \LaTeX n'en prend qu'un seul en compte.

II Explication rapide du code source

A Structure du code source

Le code source est divisé en plusieurs parties :

La définition du document : la première ligne permet de déterminer quel type de document est réalisé : on parle de la *classe* du document (d'où le nom `documentclass`).

Ici, il s'agit d'un document de type *article*. Les *options globales* sont également déterminées à ce moment : elles sont valables pour tout le document sauf indication contraire.

Le préambule : il s'agit des lignes entre `\documentclass` et `\begin{document}` (lignes 2 à 8).

Le préambule contient tous les package(s) utilisés ainsi que différentes *commandes* définies par l'utilisateur ou spécifiquement utilisées dans le préambule.

Le corps du document : situé entre `\begin{document}` et `\end{document}`, il s'agit du contenu même du document qui sera alors formaté en fonction du contenu du préambule et des commandes utilisées dans le texte.

Après : les lignes après `\end{document}` ne sont pas interprétées par \LaTeX et peuvent donc contenir ce que l'on veut : des commentaires, des notes, des parties mises de côté...

B Explication du préambule

`\documentclass[12pt,french]{article}` : cette commande indique que le document est de classe *article* et sera donc assez court. Il existe, en comparaison, la classe *book* pour écrire des documents plus longs. Bien d'autres classes existent (*letter*, *beamer*,...).

Ce document respectera la typographie française et la taille des fontes sera de 12pt (10pt étant la taille par défaut).

`\usepackage[utf8x]{inputenc}` : cette commande permet de charger le package `inputenc` avec l'option `utf8x`. Nous n'expliquerons rien en détails ici mais cela gère le *codage* d'entrée des caractères du *code source* (d'où l'intérêt d'avoir configuré *Texmaker* en `utf8`).

`\usepackage[T1]{fontenc}` : cette commande permet de charger le package `fontenc` avec l'option `T1` permettant de gérer, entre autre, les caractères accentués et notamment les « copiés-collés » à partir de fichiers PDF.

`\usepackage[upright]{fourier}` : charge le package `fourier` avec son option `upright` qui permet d'avoir les majuscules droites dans les formules mathématiques.

`\usepackage[a4paper,margin=2cm]{geometry}` : charge le package `geometry` avec différentes options de mise en page.

`\usepackage{mathtools,amssymb}` : charge le package `mathtools` qui est essentiel dans un document destiné à composer des textes scientifiques, avec un formalisme et donc une mise en page particulière. Le package `amssymb` regroupe quant à lui quantité de symboles utilisés notamment en mathématiques et en physiques.

`\usepackage{babel}` : obligatoirement le dernier de la liste. Le package `babel` permet d'assurer au rédacteur que le texte sera composé en respectant les usages propres à la langue de composition du document (ici en français). La langue peut être spécifiée en option de ce package en écrivant `\usepackage[french]{babel}` mais il est préférable d'indiquer l'option de langue avec la *classe* du document (comme nous l'avons fait). Ainsi, cette langue sera utilisée de façon globale par tous les package(s) en ayant besoin.

III Commandes

A Arguments d'une commande

Les *commandes* permettent de structurer et de mettre en forme le document. Elles sont reconnaissables car elles commencent par le caractère `\` suivi du nom de la commande (plus ou moins explicite). La commande se termine par tout caractère autre qu'une lettre (accolade, crochet, chiffre, espace, ponctuation...).

Différents types de commandes existent :

Sans paramètres : elles exécutent simplement une action : `\neq`, `\par`, `\Leftrightarrow`

Avec paramètres : il existe alors deux types de *paramètres*. Ils peuvent être :

Obligatoires : ceux-là sont notés entre accolades et il peut y avoir plusieurs paramètres par commande (une paire d'accolades pour chacun d'entre eux). Par exemple, la commande `\textbf` ne possède qu'un paramètre alors que la commande `\frac` en possède deux. Il a été dit que la commande s'arrêtait par tout caractère autre qu'une lettre.

Dans ce cas, il n'est pas toujours nécessaire d'utiliser les accolades.

Ainsi, `\frac{2}{3}` \Leftrightarrow `\frac23` \Leftrightarrow `\frac 2 3` : cela donne la fraction $\frac{2}{3}$.

En revanche, écrire `\fracab` ne donnera pas $\frac{a}{b}$, car \LaTeX ne saura pas interpréter la commande `\fracab` (à moins que vous ayez créé une commande portant ce nom).

Facultatifs : ceux-là sont notés entre crochets avant le premier argument obligatoire (qui n'existe pas toujours d'ailleurs). Les *arguments* facultatifs ou *optionnels* permettent de modifier localement l'action d'une commande. Par exemple `\sqrt[3]{x}`.

En résumé, une commande peut avoir une des trois formes suivantes :

`\commande`

`\commande{<argument 1>}{<argument 2>}...`

`\commande[<argument optionnel>]{<argument 1>}{<argument 2>}...`



Les majuscules dans les noms de commandes sont importantes : ainsi `\frac` n'est pas identique à `\Frac`.

B

Commandes semi-globales

Les *commandes locales* permettent de modifier l'aspect du texte de façon locale.

Les *commandes semi-globales* n'ont pas d'argument et modifient tout le texte qui suit jusqu'à ce qu'une autre commande semi-globale ou qu'une commande locale ne modifie encore la mise en forme. On peut limiter l'action des commandes semi-globales à l'aide d'une paire d'accolades englobant le texte mais également la commande. Autrement dit :

Commande locale : `\commande{<du texte>}`

Commande semi-globale : `{\commande <du texte>}`

La plupart du temps, on utilise les commandes locales sur des textes courts sans changement de paragraphes alors que les commandes semi-globales sont appliquées à des textes plus longs et acceptent les changements de paragraphes. Voici un exemple :

Voici un **exemple** : tout va bien mais on peut vouloir continuer avec un texte

Cela est important !
Comprenez-vous ? C'est bien !

```
1 Voici un \textbf{exemple :}  
2 tout va bien mais on peut vouloir  
3 continuer avec un texte  
4 \tiny plus petit jusqu'au bout.\par  
5 {\bfseries  
6     Cela est important !\par  
7     Comprenez-vous ?  
8 }  
9 C'est bien !
```


C Les packages

Les package(s) sont chargés à l'aide de la commande `\usepackage`. Chaque package est une *extension* de \LaTeX et c'est la création de ces milliers de package(s) qui fait que \LaTeX évolue de jours en jours. Le nom du package est l'argument obligatoire et les options sont spécifiées entre crochets si nécessaire :

$$\text{\textbackslash usepackage}[<\text{options}>]\{<\text{nom du package}>\}$$

Si plusieurs package(s) doivent être appelés sans option particulière (ou avec la même option), alors on peut les lister au sein de la même commande `\usepackage`. C'est le cas par exemple de la ligne 4 : `\usepackage{mathtools,amssymb}` fait appel à deux package(s) liés aux mathématiques.



Rappel : Le package `babel` est le package qui permet la gestion de la langue dans laquelle est écrite le document. C'est d'ailleurs la fonctionnalité de `french` signalée dans la *commande* `\documentclass`. Le package `babel` doit être, en général, le dernier de la liste des package(s) utilisés.

IV Les caractères spéciaux

On a vu que certains caractères avaient une utilisation spécifique dans le code source : c'est le cas par exemple des caractères `%` et `$` qui permettent respectivement d'entrer un commentaire et une formule mathématique. Comment faire cependant pour écrire `-20%` sur une veste à `50$` ou bien `S = \{1;3\}` ?

Le « texte » ci-dessous donne les caractères spéciaux réservés par \LaTeX et la syntaxe nécessaire pour les utiliser dans un texte classique :

Voici quelques exemples de caractères spéciaux, réservés par \LaTeX :

`\{` et `\}`, utiles pour les ensembles, donnent les accolades ouvrante et fermante,

`%` qui sert à écrire des commentaires dans le code source

`$` qui sert à ouvrir ou fermer le mode « mathématiques »

`^` qui sert pour les exposants

`_` qui sert pour les indices

`&` qui sert pour les tableaux (séparateur de colonnes)

`#` qui sert lorsque l'on définit des commandes personnelles

`~` qui produit un espace insécable



Le caractère `@` n'est pas un caractère spécial et permet d'obtenir simplement `@`. Cependant, dans certains cas, il est utilisé de façon particulière : pour les tableaux, les commandes personnelles...

Mise en forme

I Mise en forme de base

A Un premier exemple

Recopier et compiler le *code source* suivant :

```
_____ Quelques mises en forme _____  
1 \documentclass[12pt,french]{article}  
2 \usepackage[utf8x]{inputenc}  
3 \usepackage[T1]{fontenc}  
4 \usepackage[upright]{fourier}  
5 \usepackage{mathtools,amssymb}  
6 \usepackage{babel}  
7  
8  
9 \begin{document}  
10 Les consignes suivantes sont très \textbf{importantes.}\par  
11 Lisez-les avec attention :\par  
12 {\bfseries  
13 Toutes les étapes de calculs doivent être détaillées.\par  
14 Un résultat non justifié ne rapporte aucun point.  
15 }  
16  
17 Le \fbox{barème} n'est donné qu'à titre \textit{indicatif}.  
18  
19 Voilà une \underline{citation} de \textsc{César} : {\itshape Alea jacta est}.  
20 \end{document}
```

1. Nommer cinq différentes mises en forme utilisées dans ce document.
2. Quelle(s) commandes permettent d'obtenir ces mises en forme?
3. Quelle(s) différence(s) y a-t-il entre les deux commandes qui permettent de mettre du texte en gras?

B Police et fontes

Une *police* se déclinent en trois caractéristiques : famille, formes et graisses qui constituent alors un ensemble de *fontes* de cette police. Le tableau ci-dessous résume les commandes permettant d'utiliser une de ces fontes.



Le caractère □ indique qu'il faut laisser un espace dans le *code source*.

		Portée		Signification
		locale	semi-globale	des radicaux
Familles	romain (par défaut)	<code>\textrm{<texte>}</code>	<code>\rmfamily_<texte></code>	rm = roman
	sans empattement	<code>\textsf{<texte>}</code>	<code>\sffamily_<texte></code>	sf = sans serif
	à chasse fixe	<code>\texttt{<texte>}</code>	<code>\ttfamily_<texte></code>	tt = teletype
Formes	droit (par défaut)	<code>\textup{<texte>}</code>	<code>\upshape_<texte></code>	up = upright (droit)
	<i>incliné</i>	<code>\textsl{<texte>}</code>	<code>\slshape_<texte></code>	sl = slanted (penché)
	<i>italique</i>	<code>\textit{<texte>}</code>	<code>\itshape_<texte></code>	it = italique
	PETITES CAPITALES	<code>\textsc{<texte>}</code>	<code>\scshape_<texte></code>	sc = small caps
Graisses	médium (par défaut)	<code>\textmd{<texte>}</code>	<code>\mdseries_<texte></code>	ms = medium
	gras	<code>\textbf{<texte>}</code>	<code>\bfseries_<texte></code>	bf = bold face (gras)

Plusieurs commandes peuvent être utilisées conjointement.

Par exemple pour obtenir du texte en ***gras, italique et sans empattement***, on écrira :



```
\textsf{\textbf{\textit{<texte>}}}
```

L'ordre des commandes n'a pas d'importance mais il faut faire attention à avoir le bon nombre de paires d'accolades.

Changement de la taille des fontes

La taille des fontes peut être fixée de manière absolue dans le préambule, en option à `\documentclass`.

Les options disponibles sont 10pt (valeur par défaut si rien n'est indiqué), 11pt et 12pt.

Une fois définie cette taille absolue, on peut agrandir et réduire la taille d'une partie du document en utilisant des *commandes semi-globales* qui modifient alors le texte de façon relative. Le changement dépendra en effet de la taille absolue. Ces commandes sont les suivantes :

Commande	Signification et test
<code>\tiny_<text></code>	minuscule
<code>\scriptsize_<text></code>	taille des indices et exposants
<code>\footnotesize_<text></code>	Taille des notes de bas de pages
<code>\small_<text></code>	petit
<code>\normalsize_<text></code>	taille définie par l'option absolue
<code>\large_<text></code>	grand
<code>\Large_<text></code>	plus grand
<code>\LARGE_<text></code>	encore plus grand
<code>\huge_<text></code>	énorme
<code>\Huge_<text></code>	encore plus énorme



Les majuscules dans le nom des commandes sont importantes.
De plus, il s'agit de commandes *semi-globales* donc il faut penser à mettre des accolades englobantes si on veut modifier la taille d'une partie du texte seulement.

D Alignement

Par défaut, le texte est *justifié*. Cela signifie que \LaTeX gère les espaces entre les mots pour que le texte soit aligné à gauche **et** à droite.

Cependant, on peut parfois avoir besoin de centrer le texte, ou bien de demander uniquement un alignement à gauche ou uniquement un alignement à droite. Pour cela, on utilise respectivement les *environnements* `center`, `flushleft`, `flushright`.

Du texte au centre.

Alignement sur la gauche.

Alignement sur la droite.

```

1 \begin{center}
2 Du texte au centre.
3 \end{center}
4 \begin{flushleft}
5 Alignement sur la gauche.
6 \end{flushleft}
7 \begin{flushright}
8 Alignement sur la droite.
9 \end{flushright}
```



`\centering` est la commande semi-globale associée à l'environnement `center`. `\raggedleft` est associé à `flushright` et `\raggedright` est associée à `flushleft`.

E Espaces



Les espaces écrits dans le *code source* ne sont pas identiquement restitués dans le document final après compilation.

Pour cela, on parlera d'**un** espace dans le *fichier source* et d'**une** espace dans le document final.

1 Espaces horizontales

Nous l'avons vu précédemment, pour obtenir une espace entre deux mots, il suffit de saisir un espace dans le *code source* à l'aide de la barre d'espace du clavier. Cependant, saisir plusieurs espaces ne changera rien et lors de la compilation, ils seront interprétés comme un seul et même espace. De même pour un changement de ligne (sans ligne vide !) :

Du texte sur une seule ligne.

```
1 Du_texte
2      sur_une
3 seule_ligne.
```



Le symbole `~` permet d'obtenir une *espace insécable*. En fin de ligne notamment, il faudra donc écrire `Louis~XIV` si on veut éviter que Louis soit inscrit en bout de ligne et XIV au début de la ligne suivante.

Parfois, on peut avoir besoin d'une espace horizontale ayant une longueur bien précise. Cela est possible à l'aide de la commande `\hspace{<longueur>}`. L'argument `<longueur>` est spécifié à l'aide d'un nombre suivi de son unité (sans espace entre les deux). L'unité peut être `cm`, `mm` ou bien encore `pt` mais bien d'autres aussi.

De plus, la commande `\hfill` est un espace élastique. Voilà une façon de se servir de ces deux commandes :

Les consignes sont vraiment importantes.

Les consignes sont vitales!

~~Inutile~~

Exercice 1

(2 points)

```
1 Les consignes sont vraiment importantes.\par
2 Les consignes sont \hspace{1.4cm} vitales !\par
3 Inutile \hspace{-1.3cm} xxxxxx\par
4 \textbf{Exercice 1} \hfill \textit{(2 points)}
```

2 Espaces verticales

Nous l'avons vu précédemment, pour changer de paragraphe, il suffit de saisir laisser une ligne vide dans le *code source*. La commande `\par` assure la même fonction. Cependant, plusieurs lignes vides seront toujours interprétées comme un seul changement de paragraphe, de même que la succession de plusieurs commandes `\par`.

Comment faire alors apparaître dans le document final des espaces entre deux paragraphes ?

La commande `\vspace{<longueur>}` est une solution et cela fonctionne comme pour les espaces horizontales. Cependant, les commandes `\smallskip`, `\medskip` et `\bigskip` sont simples et rapides à utiliser.

Consigne importante.

Espace standard.

Petite espace.

Espace moyenne.

Grande espace.

Espace personnelle.

```
1 Consigne importante.
2
3 Espace standard.\smallskip
4
5
6
7 Petite espace.\medskip
8
9 Espace moyenne.\bigskip
10
11 Grande espace.\par\vspace{2em}
12 Espace personnelle.
```

Pour la commande `\hspace{<longueur>}`, la longueur est un nombre avec une unité accolée, parmi :

- ★ mm (millimètre) ;
- ★ cm (centimètre) ;
- ★ pt (point anglo-saxon) ;
- ★ dd (point Didot) ;
- ★ ex (hauteur d'x) ;
- ★ em (cadratin).

Les unités ex et em sont proportionnelles au corps de la police :

- ★ la hauteur d'x, parfois appelée à tort « hauteur d'œil », est la hauteur d'un bas de casse (lettre minuscule) sans hampe ni jambage, comme le x ;
- ★ le cadratin est égal au corps de la police.

Par exemple :

- ★ `\hspace{1cm}` pour une espace de un centimètre ;
- ★ `\hspace{0.5em}` pour une espace d'un demi cadratin.



La commande `\vfill` permet de créer une espace verticale élastique. Essayer de compiler l'exemple ci-dessous.

-
- 1 Le devoir est sur 20 points.\par\vfill
 - 2 Tourner la page.
-

II Couleur



Difficile de parler de couleurs sur des photocopies noir et blanc donc pensez à taper et compiler les exemples proposés.

Afin de colorer un document, on utilise le *package* `xcolor`, chargé dans le préambule à l'aide de la commande `\usepackage{xcolor}`. `xcolor` permet d'accéder aux couleurs suivantes :

```
red      magenta  gray      white   violet   olive
blue     cyan     lightgray black   purple   teal    brown
green    yellow   darkgray  orange  pink     lime
```

Là encore, il existe une commande locale et une commande semi-globale dont voilà un exemple :

Les consignes suivantes sont **importantes**.

Lisez-les avec *attention*.

Sinon, gare **À VOUS** !

```
1 \color{blue}
2 Les consignes suivantes sont
3 \textcolor{red}{importantes.}\par
4 Lisez-les avec \textit{attention}.\par
5 Sinon, gare
6 \textcolor{purple}{\textbf{\textsc{'a vous}}}
```



Le *package* `xcolor` possède différentes options qui permettent d'accéder à bien d'autres couleurs. C'est le cas de l'option `dvipsnames` qui donne accès à 68 couleurs en plus de celles de base. On écrira alors : `\usepackage[dvipsnames]{xcolor}` dans le préambule. La documentation du *package* permet d'en savoir davantage. Il suffit de taper sur un moteur de recherche LaTeX `xcolor doc` pour obtenir ce que l'on cherche.

Voilà un autre exemple qui montre comment faire des encadrements colorés.

Chapitre 1 :

L'art de faire des encadrements

I. Partie 1

```
1 \begin{center}
2   \colorbox{yellow}{\textbf{Chapitre 1 :}}\par
3   \textit{L'art de faire des encadrements}
4 \end{center}
5 \fcolorbox{red}{lightgray}{\textbf{I. Partie 1}}
```

Pour finir sur ce thème, voici la commande `\pagecolor{<couleur>}` qui permet de colorer le fond d'une page. Très utile pour créer un document destiné à être vidéoprojeté. En effet, le fond blanc d'un document projeté peut être fatigant pour les yeux des lecteurs. Allez-y : essayez !

III Mise en page

A Dimensions de la page

Par défaut, les dimensions de la page sont réglées en fonction de la *classe* du document.

Le *package* `geometry` est utilisé pour régler la géométrie de la page indépendamment du choix de la *classe* : dimensions du papier, orientation (portrait, paysage), dimensions des marges, particularités d'un document recto-verso, dimensions des en-têtes et pieds-de-pages...

Pour cela, on peut charger le *package* avec toute une liste d'options séparées par une virgule :

```
\usepackage[a4paper,margin=2cm]{geometry}.
```

Il est également possible de charger le *package* tout seul puis d'utiliser la commande `\geometry` qui prend en argument la même liste d'options. Ainsi, on peut également écrire :

```
\usepackage{a4paper,geometry}  
\geometry{margin=2cm}
```

La documentation du *package* `geometry` liste l'ensemble des options disponibles dont voici les plus courantes (<dim> est un nombre avec une unité de longueur) :

- `landscape` : orientation paysage ;
- `twoside` : document recto-verso ;
- `width=<dim>` et `height=<dim>` : largeur et hauteur de la page. On peut aussi utiliser `a4paper` ou `a5paper` (formats disponibles de A0 jusqu'à A6) ;
- `textwidth=<dim>` et `textheight=<dim>` : largeur et hauteur attribuée au texte ;
- `lmargin=<dim>` et `rmargin=<dim>` : respectivement marges intérieures (ou gauche) et extérieures (ou droite) ;
- `tmargin=<dim>` et `bmargin=<dim>` : respectivement marges de tête (t comme top) et de pied (b comme bottom) ;
- `margin=<dim>` : fixe les quatre marges précédentes avec la même longueur.

B Multicolonnes

Pour écrire une partie d'un document sur deux ou plusieurs colonnes, on a recourt au *package* `multicol` qui nous permet alors d'accéder à l'*environnement* `multicols`.



Attention, le nom du *package* ne prend pas de S final alors que le nom de l'environnement en prend un.

Voilà deux exemples d'utilisation :

— Multicolonges —

```
1 \setlength{\columnseprule}{0.4pt}
2 \begin{multicols}{2}\raggedcolumns
3   Les policiers semblent avoir mis
4   la main sur les suspects qui ne
5   courraient visiblement pas
6   assez vite.
7 \end{multicols}
```

donne :

Les policiers semblent avoir mis la main sur les suspects qui ne	courraient visiblement pas assez vite.
--	---

— Multicolonges —

```
1 \setlength{\columnseprule}{0.4pt}
2 \begin{multicols}{3}[\textbf{Formation}]
3 \raggedcolumns
4   Ce stage \LaTeX{} est incroyable.
5   Le prochain a lieu quel jour ? Ce
6   document est super !
7 \end{multicols}
```

donne :

Formation

Ce stage \LaTeX{} est incroyable. Le	prochain a lieu quel jour ? Ce do-	cument est super!
--------------------------------------	------------------------------------	-------------------



Pour changer de colonne à un point précis, on peut utiliser la commande `\columnbreak`.

Remarque :

Pour supprimer la barre verticale de séparation, il suffit de régler sa largeur à 0 mm.

```
1 \setlength{\columnseprule}{0mm}
2 \begin{multicols}{3}\raggedcolumns
3 Ce stage \LaTeX{} est incroyable.
4 \columnbreak
5 Le prochain a lieu quel jour ?
6 \columnbreak
7 Ce document est super !
8 \end{multicols}
```

donne :

Ce stage \LaTeX{} est incroyable. Le prochain a lieu quel jour ? Ce document est super !



La commande `\raggedcolumns` sert à avoir un texte aligné en haut de colonne dans chaque colonne.

IV Structurer un document

A Listes structurées

LaTeX gère par défaut trois types de listes :

- les listes d'énumération avec une liste d'*item* comme celle que vous êtes en train de lire ;
- les listes numérotées dont chaque élément est numéroté ;
- les listes de description dont chaque élément est introduit par l'objet que l'on souhaite décrire.

Voilà ce que donne la liste précédente avec les deux autres types de listes :

1. les listes d'énumération avec une liste d'*item* ;
2. les listes numérotées dont chaque élément est numéroté comme celle que vous êtes en train de lire ;
3. les listes de description dont chaque élément est introduit par l'objet que l'on souhaite décrire.

les listes d'énumération avec une liste d'*item* ;

les listes numérotées dont chaque élément est numéroté ;

les listes de description dont chaque élément est introduit par l'objet que l'on souhaite décrire comme celle que vous êtes en train de lire.

Toutes ces listes peuvent s'imbriquer les unes dans les autres en mélangeant ou non les différents types. On utilisera avantageusement les tabulations pour une présentation claire du *code source*.

- du pain ;
- du beurre ;
- de la confiture.

```
1 \begin{itemize}
2   \item du pain ;
3   \item du beurre ;
4   \item de la confiture.
5 \end{itemize}
```

1. Qu'est ce qu'un polygone ?
2. Qu'est ce qu'un parallélogramme ?
 - a) Qu'est ce qu'un rectangle ?
 - b) Qu'est ce qu'un losange ?

```
1 \begin{enumerate}
2   \item Qu'est ce qu'un polygone ?
3   \item Qu'est ce qu'un parallélogramme ?
4     \begin{enumerate}
5       \item Qu'est ce qu'un rectangle ?
6       \item Qu'est ce qu'un losange ?
7     \end{enumerate}
8 \end{enumerate}
```

Rectangle : voici un long texte dans lequel on parle du rectangle.

Losange : voici un long texte dans lequel on parle du losange.

On remarque la mise en page automatique de ce type de liste au niveau des espaces.

```
1 \begin{description}
2   \item[Rectangle :] voici un long texte dans
3     lequel on parle du rectangle.
4   \item[Losange :] voici un long texte dans lequel
5     on parle du losange.
6 \end{description}
7 On remarque la mise en page automatique
8 de ce type de liste au niveau des espaces.
```



Le *package* `enumitem` permet de personnaliser la présentation des ces différents types de listes mais également de créer de nouvelles listes.

De plus, il permet de reprendre la numérotation d'une liste `enumerate` qui a été interrompue. La lecture de la documentation de ce *package* est vivement conseillée.

B Sectionnement

Les commandes de *sectionnement* permettent d'établir le plan du document. Les commandes les plus fréquemment utilisées sont :

```
\part[<titre court>]{<Titre>}
\chapter[<titre court>]{<Titre>}
\section[<titre court>]{<Titre>}
\subsection[<titre court>]{<Titre>}
\subsubsection[<titre court>]{<Titre>}
```



La commande `\chapter` n'existe pas dans la classe `article`.

Le `<titre court>` est optionnel et permet d'afficher un titre différent dans la table des matières ou dans les en-têtes.

Recopier le code suivant et observer le résultat de la compilation :

Commandes de sectionnement

```
1 \documentclass[12pt,french]{book}
2 \usepackage[utf8x]{inputenc}
3 \usepackage[T1]{fontenc}
4 \usepackage[upright]{fourier}
5 \usepackage[a4paper,margin=2cm]{geometry}
6 \usepackage{mathtools,amssymb}
7 \usepackage{babel}
8
9 \begin{document}
```

```

10 \part{Analyse}
11
12 \chapter{Les suites}
13
14 \section{Introduction}
15
16 \section{Limite d'une suite}
17 \subsection{Limite infinie}
18 \subsection{Limite finie}
19
20 \section{Suites géométriques}
21 \subsection{Limite}
22 \subsection{Somme des termes}
23 \end{document}

```



Si on ne souhaite pas de numérotation à un endroit, on peut ajouter `*`. On écrira par exemple : `\section*{Introduction}` à la place de `\section{Introduction}`.

On peut modifier la mise en forme de la numérotation des commandes `\section` et `\subsection` de la façon suivante :

```

1 \renewcommand{\thesection}{\Roman{section}.}
2 \renewcommand{\thesubsection}{\Alph{subsection}.}

```

Il suffit de taper les deux lignes précédentes dans le préambule et de recompiler. Faites un essai et essayer de comprendre le fonctionnement des commandes utilisées.



Pour modifier avec plus de précision les différents types de sectionnement, on pourra lire attentivement les documentations des *packages* `sectsty` et `titlesec`.



Références croisées

Les *renvois* sont gérés automatiquement par \LaTeX . L'intérêt de cela est évidemment de pouvoir modifier à loisir son document sans être obligé de se demander si telle ou telle référence a été changée de place ou de numérotation. Cela est très utile pour se reporter à une section ou bien encore à une liste.

Voilà un exemple :

V Renvois

1. Qui suis-je ?

2. Qui est-il ?

Nous voyons que la question 2. n'est pas si simple (section V page 24)

```
1 \section{Renvois}\label{sec}
2   \begin{enumerate}
3     \item Qui suis-je ?\label{qu:je}
4     \item Qui est-il ?\label{qu:il}
5   \end{enumerate}
6 Nous voyons que la question \ref{qu:il}
7 n'est pas si simple
8 (section \ref{sec} page \pageref{sec})
```

La commande `\label{<étiquette>}` permet d'apposer une étiquette à l'élément que l'on souhaite référencer (ou qui est susceptible de l'être). Dans l'idéal, il faut essayer de choisir des étiquettes avec un nom suffisamment évocateur (ce qui n'est pas le cas de l'exemple précédent).

La commande `\ref{<étiquette>}` se place à l'endroit même où l'on souhaite faire notre référence à l'élément précédemment étiqueté. Le numéro alors affiché correspond au numéro de l'élément. Enfin, `\pageref{<étiquette>}` indique le numéro de la page à laquelle se trouve l'élément à référencer.



Pour utiliser les références croisées, une seule compilation ne suffit pas. En effet, la première compilation permet simplement d'enregistrer les différentes étiquettes dans un fichier auxiliaire. Afin de pouvoir ensuite référencer ces étiquettes dans le texte, il faut procéder à une deuxième compilation. Sinon, on verra apparaître le symbole `??`.

Mathématiques



Pour écrire des mathématiques avec \LaTeX , il faut accéder au mode « mathématiques » à l'aide du caractère $\$$. Ce même caractère est utilisé pour sortir du mode « mathématiques ».

I Un premier exemple

Voilà le préambule que nous utiliserons dans cette section :

Préambule

```
1 \documentclass[10pt,french]{article}
2 \usepackage[utf8x]{inputenc}
3 \usepackage[T1]{fontenc}
4 \usepackage[a4paper,margin=1.5cm]{geometry}
5 \usepackage[dvipsnames]{xcolor}
6 \usepackage{mathtools,amssymb}
7 \usepackage[upright]{fourier}
8 \usepackage[autolanguage,np]{numprint}
9 \usepackage{xlop}
10 \usepackage{cancel}
11 \renewcommand\CancelColor{\color{red}}
12 \usepackage{dsfont}
13 \usepackage{babel}
14 \DecimalMathComma
```

Nous avons déjà signalé que les *package* `amssymb` et `amstools` servaient de « boîte à outils » pour les mathématiques. Parmi des nombreuses commandes, ces *packages* permettent notamment d'écrire les opérations arithmétiques basiques.

$$1 + 2 = 3 \text{ et } 3 \neq 4$$

$$(-5) - (-8) = 3$$

$$9 \times 7 = 63$$

$$25 \div 6 \approx 4,17$$

$$1 \ \$1 + 2 = 3\$ \text{ et } \$3 \neq 4\$ \backslash \text{par}$$

$$2 \ \$(-5) - (-8) = 3\$ \backslash \text{par}$$

$$3 \ \$9 \times 7 = 63\$ \backslash \text{par}$$

$$4 \ \$25 \div 6 \approx 4,17\$$$



Dans le préambule, la commande `\DecimalMathComma` permet de définir la virgule comme séparateur entre la partie entière et la partie décimale d'un nombre. Cela permet d'éviter la création d'espaces non souhaitée dans l'écriture des nombres en français.

D'autre part, certains *packages* viennent compléter les possibilités déjà nombreuses.



Les *codes sources* suivants peuvent s'écrire à la suite du préambule présenté ci-dessus mais il ne faut alors pas oublier de les encadrer avec `\begin{document}` et `\end{document}`.

A Utilisation de `\usepackage[autolanguage,np]{numprint}`

En français, les grands nombres doivent être écrits en séparant les chiffres par tranche de 3 en utilisant des espaces fines. Le *package* `numprint` rend cela possible à l'aide de la commande `\np`. Celle-ci est un raccourci créé grâce à l'option `np` du *package*. La vraie commande se nomme `\numprint`. De plus, cette commande peut prendre en option une unité de mesure, gérant ainsi automatiquement les espaces.

10 m = 10 000 mm = 0,01 km
10 km/h = 10 km.h⁻¹ ≈ 2,78 m/s

```
1 $\np[m]{10} = \np[mm]{10000} = \np[km]{0,01}$  
2  
3 $\np[km/h]{10} = \np[km.h^{-1}]{10}$  
4 \approx \np[m/s]{2,78}$
```



L'option `autolanguage` permet simplement à `numprint` de s'accorder avec les règles de la langue en cours (notamment pour le séparateur de milliers : espace, point ou virgule).

Ce *package* permet aussi d'écrire `$\np{1,54e-3}$` qui donne alors $1,54 \cdot 10^{-3}$. La documentation du *package* renseigne sur les différentes options possibles et les modifications envisagées.

B Utilisation de `\usepackage{xlop}`

Le *package* `xlop` permet d'écrire les commandes du code ci-dessous et dispose de nombreuses options. Il ne faut pas hésiter à consulter sa documentation.

Les opérations posées

```
1 \opset{decimalsepsymbol={,},voperator=bottom,voperation=top}  
2  
3 \opadd{45,05}{78,4}\quad ou encore \opadd[style=text]{45.05}{78.4}\medskip  
4  
5 \opsub[carrysub,lastcarry,columnwidth=2.5ex,offsetcarry=-0.4,decimalsepoffset=-3pt,deletezero=false]  
6 {12.34}{5.67} \quad ou encore \opsub[style=text]{12.34}{5.67}\medskip  
7  
8 \opmul[lineheight=20pt]{35684}{7.9}  
9 \quad  
10 ou encore \opmul[style=text]{35684}{7.9}\medskip  
11
```

```

12 \opdiv{25}{7} \quad ou encore \opdiv[style=text]{25}{7} \medskip
13
14 \opdiv[maxdivstep=3]{25}{7} \quad ou encore \opdiv[style=text,maxdivstep=3]{25}{7}

```

Voici le résultat de ce code :

$$\begin{array}{r}
 1 \\
 45,05 \\
 + 78,4 \\
 \hline
 123,45
 \end{array}
 \quad \text{ou encore } 45,05 + 78,4 = 123,45$$

$$\begin{array}{r}
 12,34 \\
 - 5,67 \\
 \hline
 6,67
 \end{array}
 \quad \text{ou encore } 12,34 - 5,67 = 6,67$$

$$\begin{array}{r}
 35684 \\
 \times 7,9 \\
 \hline
 281156 \\
 249788 \\
 \hline
 281903,6
 \end{array}
 \quad \text{ou encore } 35684 \times 7,9 = 281903,6$$

$$\begin{array}{r}
 25 \overline{) 7} \\
 4 \overline{) 3}
 \end{array}
 \quad \text{ou encore } 25 = 7 \times 3 + 4$$

$$\begin{array}{r}
 25 \overline{) 7} \\
 40 \overline{) 3,57} \\
 50 \\
 1
 \end{array}
 \quad \text{ou encore } 25 \div 7 \approx 3,57$$

C Utilisation de `\usepackage{cancel}`

L'exemple suivant montre comment utiliser simplement le *package* `cancel` qui définit la commande `\cancel`. Dans le préambule, la ligne `\renewcommand\CancelColor{\color{red}}` permet de définir la couleur du trait utilisé par `\cancel` :

$$\frac{a \times \cancel{c}}{\cancel{c} \times b} = \frac{a}{b}$$

$$\cancel{2x^2} - 2x + 4 - \cancel{2x^2} + 3x = x + 4$$

```

1 $\dfrac{a \times \cancel{c}}{\cancel{c} \times b} = \dfrac{a}{b}$
2 = \dfrac a b$\par\medskip
3 $\cancel{2x^2}-2x+4-\cancel{2x^2}+3x=x+4$

```

D Utilisation `\usepackage{dsfont}`

Le dernier *package* utilisé dans notre préambule est `dsfont` et permet simplement d'écrire les ensembles mathématiques à l'aide de la commande `\mathds`. Profitons en pour donner quelques symboles liés aux ensembles (inclusion, appartenance...).

Voici un exemple :

```

1 $\varnothing = \emptyset \subset \mathds N \subset \mathds Z \subset
2 \mathds D \subset \mathds Q \subset \mathds R \subset \mathds C$ \medskip
3
4 $\mathds R^* = ]-\infty ; 0[ \cup ]0 ; +\infty[ = \mathds R \setminus \{0\}$.
```

On obtient le résultat suivant :

$$\emptyset = \varnothing \subset \mathbb{N} \subset \mathbb{Z} \subset \mathbb{D} \subset \mathbb{Q} \subset \mathbb{R} \subset \mathbb{C}$$

$$\mathbb{R}^* =]-\infty; 0[\cup]0; +\infty[= \mathbb{R} \setminus \{0\}.$$

E Fontes mathématiques

Le *package* `amstools` propose la commandes `\mathbb` pour noter les ensembles à l'aide de caractères ajourés et la commande `\mathbf` pour écrire des caractères gras en mode mathématiques (la commande `\textbf` ne fonctionne pas dans ce mode). Certains préféreront l'une ou l'autre méthode pour écrire des ensembles à la place du *package* `dsfont`.

$$\mathbb{N} \subset \mathbb{Z} \subset \mathbb{D} \subset \mathbb{Q} \subset \mathbb{R} \subset \mathbb{C}$$

$$\mathbf{N} \subset \mathbf{Z} \subset \mathbf{D} \subset \mathbf{Q} \subset \mathbf{R} \subset \mathbf{C}$$

$$\mathbb{N} \subset \mathbb{Z} \subset \mathbb{D} \subset \mathbb{Q} \subset \mathbb{R} \subset \mathbb{C}$$

```

1 $\mathbb N \subset \mathbb Z \subset \mathbb D
2 \subset \mathbb Q \subset \mathbb R \subset \mathbb C$
3
4 $\mathbf N \subset \mathbf Z \subset \mathbf D
5 \subset \mathbf Q \subset \mathbf R \subset \mathbf C$
6
7 $\mathds N \subset \mathds Z \subset \mathds D
8 \subset \mathds Q \subset \mathds R \subset \mathds C$
```

De plus, la commande `\mathcal` permet d'obtenir des lettres caligraphiées : la courbe \mathcal{C} .

Le *package* `mathrsfs` fournit la commande `\mathscr` : la courbe \mathscr{C} .

II Les modes mathématiques

A En ligne ou hors du texte ?

En réalité, \LaTeX possède deux modes mathématiques : le mode *en ligne* est délimité par le caractère `$` et est utilisé lorsque des mathématiques sont écrites au sein même d'un texte.

Soit f la fonction définie pour tout nombre $x \in \mathbb{R}$ par $f(x) = \frac{1}{2}x + 2$. Cette fonction est une fonction affine croissante.

```

1 Soit $f$ la fonction d'efinie pour tout nombre
2 $x \in \mathds R$ par $f(x) = \frac{1}{2} x + 2$.
3 Cette fonction est une fonction affine croissante.
```

On constate que dans ce mode là, les mathématiques sont composées de telles sortes que les espaces interlignes restent inchangées, ce qui est typographiquement meilleur.

S'il existe un mode *en ligne* pour écrire à l'intérieur des lignes d'un texte, alors il existe un mode *hors texte*. Celui-ci est délimité par les commandes `\[` et `\]`.

Soit f la fonction définie pour tout nombre $x \in \mathbb{R}$ par

$$f(x) = \frac{1}{2}x + 2.$$

Cette fonction est une fonction affine croissante.

Dans ce cas, les mathématiques sont composées dans une nouvelle ligne centrée et la taille des symboles mathématiques est adaptée.

Cependant, il se peut que l'on ait besoin d'écrire des mathématiques dans le texte mais avec des symboles ayant leur taille hors-texte. Pour cela, on pourra utiliser la commande `\displaystyle`. L'effet inverse est obtenu avec la commande `\textstyle`.

Soit f la fonction définie pour tout nombre $x \in \mathbb{R}$ par $f(x) = \frac{1}{2}x + 2$. Cela est typographiquement incorrect. On peut noter :

$$f(x) = \frac{1}{2}x + 2$$

mais cela est bizarre.

```
1 Soit  $f$  la fonction définie pour tout nombre
2  $x \in \mathbb{R}$  par  $f(x) = \frac{1}{2}x + 2$ .
3 Cette fonction est une fonction affine croissante.
```

```
1 Soit  $f$  la fonction définie pour tout nombre
2  $x \in \mathbb{R}$  par
3  $\displaystyle f(x) = \frac{1}{2}x + 2$ .
4 Cela est typographiquement incorrect. On peut noter :
5  $\textstyle f(x) = \frac{1}{2}x + 2$ 
6 mais cela est bizarre.
```



Les changements de paragraphe (à l'aide d'une ligne vide ou de la commande `\par` ou toute autre méthode) sont rigoureusement interdits à l'intérieur des modes mathématiques.

B Du texte dans des formules

Il est courant de devoir écrire des morceaux de phrases à l'intérieur d'une ligne mathématique. Le problème est que dans n'importe quel mode mathématique, les lettres sont considérées comme des variables et sont donc formatées selon les règles typographiques en mathématique, c'est-à-dire en italique. Pour bien comprendre cela, comparer les deux lignes suivantes :

$$f(x) = \frac{1}{2}x + 2 \text{ donc } f(2) = 3$$

$$f(x) = \frac{1}{2}x + 2 \text{ donc } f(2) = 3$$

La commande `\text` est celle qui nous sauve ! Évidemment, cette commande n'a pas spécialement d'utilité en mode *en ligne* comme nous le pouvons le constater dans l'exemple ci-dessous :

$$f(x) = \frac{1}{2}x + 2 \text{ donc } f(2) = 3.$$

$$f(x) = \frac{1}{2}x + 2 \text{ donc } f(2) = 3$$

```
1  $f(x) = \frac{1}{2}x + 2$  donc  $f(2) = 3$ .
2  $f(x) = \frac{1}{2}x + 2$  \text{ donc }  $f(2) = 3$ 
```



Les espaces étant gérées de façon particulière dans les modes mathématiques, on constate que `\text{donc}` ne donne pas de résultat satisfaisant. Il faut donc ajouter les espaces autour du mot pour obtenir ce que l'on souhaite : `\text{ donc }`.

C Les espaces

Dans les modes mathématiques, les espaces saisis au clavier sont tout simplement ignorés et \LaTeX gère tout seul les calculs pour espacer les symboles mathématiques. En général, ces espaces conviennent parfaitement mais il peut s'avérer nécessaire de gérer soit même les espaces en utilisant des commandes particulières.

\square = barre d'espace simple

Commande	Nom	Résultat
<code>\!</code>	Espace fine négative	$\square\square$
\square	Espace par défaut	$\square\square$
<code>\,</code>	Espace fine	$\square\square$
<code>\:</code>	Espace moyenne	$\square\square$
<code>\;</code>	Espace épaisse	$\square\square$
<code>\square</code>	Espace inter-mot	$\square\square$
<code>\quad</code>	1 cadratin	$\square\square$
<code>\qquad</code>	2 cadratins	$\square\square$



Un cadratin est égal à 1em donc la commande `\quad` est en fait un raccourci de `\hspace{1em}`.

Reprenons des exemples d'intervalles.

$\mathbb{R}^* =]-\infty; 0[\cup]0; +\infty[= \mathbb{R} \setminus \{0\}.$

$] -\infty; 4] \cap] -2; +\infty[=] -2; 4].$

```

1 $\mathds R^* = ]-\infty ; 0[ \cup ]0 ; +\infty[ =
2 \mathds R \setminus \{0\}$. \par \medskip
3 $] -\infty ; 4] \cap ] -2 ; +\infty[ = ] -2 ; 4]$.

```

Les espaces ne sont ici guère satisfaisantes autour des crochets et autour des points-virgules. Nous pouvons alors proposer la solution suivante :

$$\mathbb{R}^* =]-\infty; 0[\cup]0; +\infty[= \mathbb{R} \setminus \{0\}.$$

$$]-\infty; 4] \cap]-2; +\infty[=]-2; 4].$$

```

1 $\mathds R^*=\; \; ]-\infty\,, \; 0[\,, \;\cup\,,
2 \;\; ]0\,, \; \; \backslash\,, \; +\infty[\backslash\;
3 =\mathds R \;\setminus\; \{0\}\$.\par\medskip
4 $\;]-\infty\,, \; \; 4]\backslash\,, \;\cap\,, \; ]-2\,, \; \; +\infty[\;\backslash\;
5 =\backslash\; \; ]-2\backslash\,, \; \; 4]\$.
```



Cela peut paraître bien lourd à gérer mais on règle cela à l'aide des *commandes personnelles*.

Voici, par exemple, mes commandes personnelles pour définir les différents intervalles :

—— Commandes personnelles - Intervalles ——

```

1 \newcommand{\intervalleff}[2]{\left[#1\,,\;#2\right]}
2 \newcommand{\intervallefo}[2]{\left[#1\,,\;#2\right[}
3 \newcommand{\intervalleof}[2]{\left[#1\backslash\,,\;#2\right]}
4 \newcommand{\intervalleoo}[2]{\left[#1\backslash\,,\;#2\right[}
```

III Écrire des mathématiques de la sixième à la terminale

A

Au collège

1

En Sixième

En **sixième**, les élèves apprennent à maîtriser les notations géométriques. Les crochets et les parenthèses s'obtiennent classiquement en appuyant sur la touche correspondante. Cependant, les symboles de droites parallèles et droites perpendiculaires s'obtiennent à l'aide d'une commande spéciale.

—— Géométrie en sixième ——

```

1 Les segments $[AB]$ et $[CD]$ ont la même longueur.
2 De plus, les droites $(AB)$ et $(CD)$ sont parallèles, on note $(AB) \backslash\!\!\! \backslash (CD)$. \par
3 Et on notera $(d) \perp (d')$ si les droites $(d)$ et $(d')$ sont perpendiculaires. \medskip
4
5 $C \in (AB)$ mais $C \notin [AB]$.
```

ce qui donne :

Les segments $[AB]$ et $[CD]$ ont la même longueur. De plus, les droites (AB) et (CD) sont parallèles, on note $(AB) \parallel (CD)$.

Et on notera $(d) \perp (d')$ si les droites (d) et (d') sont perpendiculaires.

$C \in (AB)$ mais $C \notin [AB]$.

2 En Cinquième

En **cinquième**, on travaille notamment sur les fractions. L'exemple ci-dessous nous permet de montrer que la commande `\dfrac{ }{ }` est un raccourci de `\displaystyle{\frac{ }{ }}`.

Les fractions

```
1 Si $c \neq 0$, $\frac{a}{c} + \frac{b}{c} = \frac{a+b}{c}$ et, si de plus
2 $b \neq 0$, $\dfrac{a}{c} \times \dfrac{c}{b} = \dfrac{a \times c}{c \times b}$
3 $=\dfrac{a \times \cancel{c}}{\cancel{c} \times b} = \dfrac{a}{b}$.
4
5 On peut alors calculer : $3 \times \left(\dfrac{3}{2} + 2\right) - 1$.
```

ce qui donne :

Si $c \neq 0$, $\frac{a}{c} + \frac{b}{c} = \frac{a+b}{c}$ et, si de plus $b \neq 0$, $\frac{a}{c} \times \frac{c}{b} = \frac{a \times c}{c \times b} = \frac{a \times \cancel{c}}{\cancel{c} \times b} = \frac{a}{b}$.

On peut alors calculer : $3 \times \left(\frac{3}{2} + 2\right) - 1$.

Dans l'exemple suivant, on voit bien la différence obtenue sur les parenthèses en ajoutant les commandes `\left` et `\right` :

$$3 \times \left(\frac{3}{2} + 2\right) - 1$$

$$3 \times \left(\frac{3}{2} + 2\right) - 1$$

```
1 $3 \times (\dfrac{3}{2} + 2) - 1$ \par\medskip
2 $3 \times \left(\dfrac{3}{2} + 2\right) - 1$
```



Une commande `\left{ }` se termine nécessairement par son homologue `\right{ }`. Essayez alors les combinaisons `\left{ }` et `\right{ }`, `\left[]` et `\right[]` puis `\left|` et `\right|`.

Les angles sont également bien présents dans le programme de géométrie et les commandes `\widehat{ }` et `\deg` sont alors très utiles.

Notations des angles

```
1 Dans un triangle, la somme des angles est égale à $180\deg$ :
2 $\widehat{ABC} + \widehat{BAC} + \widehat{ACB} = 180\deg$.
3 Ou encore : $\widehat{A} + \widehat{B} + \widehat{C} = 180\deg$.
```

Le *code source* précédent donne :

Dans un triangle, la somme des angles est égale à 180° : $\widehat{ABC} + \widehat{BAC} + \widehat{ACB} = 180^\circ$.

Ou encore : $\widehat{A} + \widehat{B} + \widehat{C} = 180^\circ$.



La commande `\widehat{ }` permet d'obtenir un des nombreux symboles étirables horizontalement. Nous en croiserons d'autres dans les exemples à venir.

3 En Quatrième

En classe de **quatrième**, les divisions de fractions sont apprises mais également les puissances. La mise en puissance en mode mathématique est réalisée à l'aide de la syntaxe `<maths>^<exposant>`. Par exemple `3^2` donne 3^2 . Et que donne `3^{21}` ?



On pourra écrire des puissances de puissances en faisant bien attention aux groupes entre accolades.

Voici un exemple

Fractions et puissances

```
1 $\left(\dfrac{a}{b}\right)^n = \dfrac{a^n}{b^n}$ \quad et \quad \quad
2 $\dfrac{\dfrac{a}{b}}{\dfrac{c}{d}} = \dfrac{a}{b} \times \dfrac{d}{c}$
3 \quad et \quad \quad $\left(a^m\right)^n = a^{mn}$. \medskip
4
5 $\dfrac{\frac{3}{2} + 1}{\frac{5}{2}} = \left(\dfrac{3}{2} + 1\right) \times \dfrac{2}{5}$.
```

qui donne

$$\left(\frac{a}{b}\right)^n = \frac{a^n}{b^n} \quad \text{et} \quad \frac{\frac{a}{b}}{\frac{c}{d}} = \frac{a}{b} \times \frac{d}{c} \quad \text{et} \quad (a^m)^n = a^{mn}.$$

$$\frac{\frac{3}{2} + 1}{\frac{5}{2}} = \left(\frac{3}{2} + 1\right) \times \frac{2}{5}.$$

Profitons de parler des exposants pour indiquer la syntaxe et un exemple pour les indices : `<maths>_<indice>`.

L'exemple suivant donne une autre utilisation des exposants et des indices ainsi qu'un nouveau symbole étirable horizontalement : l'accolade. On notera également l'utilisation des commandes `\np` et `\text`.

Accolades horizontales

```
1 $10^n = \overbrace{10 \times 10 \times \dots \times 10}^{n \text{ fois}} \quad \text{\text{fois}}
2 = 1 \underbrace{00 \dots 0}_n \quad \text{\text{zéros}}$. \par \medskip
3 Exemple : $10^{12} = \np{1000000000000}$. \medskip
```

Ce code source donne :

$$10^n = \overbrace{10 \times 10 \times \dots \times 10}^{n \text{ fois}} = 1 \underbrace{00 \dots 0}_n.$$

Exemple : $10^{12} = 1\,000\,000\,000\,000$.



Notons que la commande `\dots` permet d'obtenir trois points de suspension mais que ceux-là sont alignés verticalement automatiquement selon le contexte dans lequel la commande est utilisée.

La résolution des équations est également un moment important dans la vie d'un collégien et bien que le symbole d'équivalence (\Leftrightarrow) ne soit pas exigible, profitons tout de même de l'occasion pour en parler tout en mettant en avant une utilisation de l'espace cadratin.

Équivalences

```

1 $3x + 2 = 5x - 1 \quad \Leftrightarrow \quad 3x - 5x = -1 - 2
2 \quad \Leftrightarrow \quad -2x = -3
3 \quad \Leftrightarrow \quad x = \dfrac{3}{2}
4
5 L'ensemble des solutions de l'équation est :
6 $\mathcal{S} = \left\{\dfrac{3}{2}\right\}.$

```

$$3x + 2 = 5x - 1 \Leftrightarrow 3x - 5x = -1 - 2 \Leftrightarrow -2x = -3 \Leftrightarrow x = \frac{3}{2}$$

L'ensemble des solutions de l'équation est : $\mathcal{S} = \left\{\frac{3}{2}\right\}$.

4 En Troisième

Les élèves de **troisième** découvrent la joie de la trigonométrie. La commande `\cos` permet simplement d'obtenir \cos

$$\cos(\widehat{ABC}) = \frac{AB}{BC} \quad \text{1} \quad \cos(\widehat{ABC}) = \frac{AB}{BC}$$

et ses acolytes `\sin` et `\tan` permettent d'obtenir \sin et \tan (caractères droits et non en italique). Pour finir, voilà un exemple à compiler pour voir différentes notations vues en classe de **troisième**.

En troisième

```

1 $\sqrt{\dfrac{a}{b}} = \dfrac{\sqrt{a}}{\sqrt{b}}$ \medskip
2
3 Si $f(x) = 3x - 2$ alors $f(-4) = 3 \times (-4) - 2 = -14$.\medskip
4
5 $3x + 2 \leq 5x - 1 \quad \Leftrightarrow \quad 3x - 5x \leq -1 - 2
6 \quad \Leftrightarrow \quad -2x \leq -3
7 \quad \Leftrightarrow \quad x \geq \dfrac{3}{2}$ \medskip
8
9 $\mathcal{V} = \dfrac{4}{3} \pi R^3$.\medskip
10
11 Si l'angle au centre $\widehat{BOA}$ intercepte le même arc $\wideparen{AB}$
12 que l'angle inscrit $\widehat{BCA}$ alors $\widehat{BOA} = 2\widehat{BCA}$.\medskip

```

ce qui donne

$$\sqrt{\frac{a}{b}} = \frac{\sqrt{a}}{\sqrt{b}}$$

Si $f(x) = 3x - 2$ alors $f(-4) = 3 \times (-4) - 2 = -14$.

$$3x + 2 \leq 5x - 1 \Leftrightarrow 3x - 5x \leq -1 - 2 \Leftrightarrow -2x \leq -3 \Leftrightarrow x \geq \frac{3}{2}$$

$$\mathcal{V} = \frac{4}{3}\pi R^3.$$

Si l'angle au centre \widehat{BOA} intercepte le même arc \widehat{AB} que l'angle inscrit \widehat{BCA} alors $\widehat{BOA} = 2\widehat{BCA}$.

B Au lycée

1 En 2nde

Nous avons déjà vu comment noter les ensembles de nombres. Cependant en classe de **seconde**, une grande importance est accordée aux fonctions. L'exemple suivant montre que l'utilisation de la commande `\colon` est largement préférée aux deux-points classiques : pour une gestion correcte des espaces par \LaTeX . La commande `\mapsto` est également à retenir.

$f: x \mapsto f(x)$ est définie sur \mathcal{D}_f .

On note \mathcal{C} sa courbe représentative.

```

1 $f\colon x \mapsto f(x)$ est d'efinie sur
2                                     $\cal D_f$.
3
4 On note $\cal C$ sa courbe repr'esentative.
```

Les vecteurs font leur apparition au début du lycée. Cela nous permet alors de découvrir un nouveau symbole étirable horizontalement — la flèche — ainsi qu'un symbole étirable verticalement — la norme d'un vecteur.

Les vecteurs

```

1 $\overrightarrow{AB} + \overrightarrow{BC} = \overrightarrow{AC}$\medskip
2
3 $\overrightarrow{AB}\binom{x_B - x_A}{y_B - y_A}$ \quad donc \quad
4 $\overrightarrow{AB}\dbinom{x_B - x_A}{y_B - y_A}$\par\medskip
5 $\left\lVert \overrightarrow{AB} \right\rVert = AB = \sqrt{(x_B - x_A)^2 + (y_B - y_A)^2}$
6
7 \medskip
8
9 $\left\lVert \lambda \overrightarrow{AB} \right\rVert =
10 \left\lVert \lambda \overrightarrow{AB} \right\rVert \times \left\lVert \overrightarrow{AB} \right\rVert$\par\medskip
11 Cela est évidemment valable dans un repère orthonormé
12 $\left(0 ; \overrightarrow{\imath}, \overrightarrow{\jmath}\right)$.
```

Ce code source donne :

$$\overrightarrow{AB} + \overrightarrow{BC} = \overrightarrow{AC}$$

$$\overrightarrow{AB} \binom{x_B - x_A}{y_B - y_A} \quad \text{donc} \quad \overrightarrow{AB} \dbinom{x_B - x_A}{y_B - y_A}$$

$$\left\| \overrightarrow{AB} \right\| = AB = \sqrt{(x_B - x_A)^2 + (y_B - y_A)^2}$$

$$\left\| \lambda \overrightarrow{AB} \right\| = |\lambda| \times \left\| \overrightarrow{AB} \right\|$$

Cela est évidemment valable dans un repère orthonormé $(O; \overrightarrow{i}, \overrightarrow{j})$.

On notera l'utilisation un peu faussée de la commande `\binom` qui sert en réalité pour l'écriture des coefficients binomiaux. Cela dit, elle est très pratique pour écrire les coordonnées d'un vecteur. Tout comme `\dfrac`, la commande `\dbinom{}{}` est un raccourci pour `\displaystyle{\binom{}{}}`.



Les commandes `\imath` et `\jmath` sont bien pratiques pour obtenir un i et un j sans point : les lettres i et j peuvent donc recevoir une flèche.

En transition avec la classe de **seconde** et celle de **première**, parlons un peu de statistiques

Moyenne : \bar{x}

Écart-type : σ

```
1 Moyenne : $\overline{x}$\par
```

```
2 \'Ecart-type : $\sigma$
```

2 En première

Pour la classe de **première**, on rencontre notamment les symboles et notations suivants :

- `Δ` permet d'obtenir Δ ;
- `\pm` permet d'obtenir \pm ;
- `\cdot` permet d'obtenir le point du produit scalaire ;
- `\hookrightarrow` permet d'obtenir \hookrightarrow .

Les vecteurs

```
1 \Equation de la tangente en $x_0$ :
2 $y = f'(x_0) \left(x - x_0\right) + f(x_0)$ avec
3 $f' = \left(\dfrac{u}{v}\right)' = \dfrac{u'v - uv'}{v^2}$.\medskip
4
5 $\Delta = b^2 - 4ac$. Si $\Delta > 0$ alors
6 $x = \dfrac{-b \pm \sqrt{\Delta}}{2a}$.\medskip
7
8 $\overrightarrow{AB} \cdot \overrightarrow{CD} = xx' + yy'$.\medskip
9
10 $u_{n+1} = q \times u_n = u_0 \times q^{n+1}$.\medskip
11
12 $1 + q + q^2 + \dots + q^n = \sum_{i=0}^n q^i$.\medskip
13
14 $\mathds{P}(X = k) = \binom{n}{k} \times p^k \times (1-p)^{n-k}$.\medskip
15
16 $X$ suit la loi binomiale de paramètres $n=10$ et $p=0,2$ :
17 $X \hookrightarrow \mathcal{B}(10 ; 0,2)$.
```

Le *code source* précédent donne :

Équation de la tangente en x_0 : $y = f'(x_0)(x - x_0) + f(x_0)$ avec $f' = \left(\frac{u}{v}\right)' = \frac{u'v - uv'}{v^2}$.

$$\Delta = b^2 - 4ac. \text{ Si } \Delta > 0 \text{ alors } x = \frac{-b \pm \sqrt{\Delta}}{2a}.$$

$$\overrightarrow{AB} \cdot \overrightarrow{CD} = xx' + yy'.$$

$$u_{n+1} = q \times u_n = u_0 \times q^{n+1}.$$

$$1 + q + q^2 + \dots + q^n = \sum_{i=0}^n q^i$$

$$\mathbb{P}(X = k) = \binom{n}{k} \times p^k \times (1 - p)^{n-k}.$$

X suit la loi binomiale de paramètres $n = 10$ et $p = 0,2$: $X \hookrightarrow \mathcal{B}(10; 0,2)$.

L'exemple suivant montre en quoi le mode « mathématiques » *en ligne* peut encore être différent du mode *hors texte* toujours dans un souci de respect des espaces inter-lignes.

_____ Somme de termes consécutifs _____

```
1 Formule :
2 $1 + q + q^2 + \dots + q^n = \sum_{i = 0}^n q^i$
3
4 \medskip
5
6 Formule :
7 \[1 + q + q^2 + \dots + q^n = \sum_{i = 0}^n q^i\]
```

Formule : $1 + q + q^2 + \dots + q^n = \sum_{i=0}^n q^i$

Formule :

$$1 + q + q^2 + \dots + q^n = \sum_{i=0}^n q^i$$

3 En Terminale

Finissons avec la classe de **terminale**.

Quelques nouvelles notations apparaissent... et du coup de nouvelles commandes \LaTeX :

La commande `\mathrm` (il s'agit d'une autre fonte mathématique pas encore rencontrée) permet d'obtenir un *d* droit en mode « mathématiques ».

Les commandes `\int_{\scriptstyle}` et `\lim_{\scriptstyle}` permettent d'écrire des intégrales et des limites.

La commande `\substack` permet d'écrire les arguments des limites sur plusieurs lignes.



Les commandes `\int_{\scriptstyle}` et `\lim_{\scriptstyle}` ne donnent pas le même résultat *en ligne* ou *hors texte*.

$$\int_0^1 x^2 dx = \left[\frac{x^3}{3} \right]_0^1 = \frac{1}{3}.$$

$$\lim_{x \rightarrow +\infty} e^x = +\infty.$$

On parle de la fonction $x \mapsto \exp(x)$.

$$\lim_{x \rightarrow 0^+} \ln(x) = -\infty \text{ ou encore}$$

$$\lim_{\substack{x \rightarrow 0 \\ x > 0}} \ln(x) = -\infty.$$

$$a \equiv b [n] \Leftrightarrow \exists k \in \mathbb{Z}, a - b = kn.$$

$$|e^{i\theta}| = 1.$$

ou

$$|e^{i\theta}| = 1.$$

```

1 $\int_0^1 x^2 \mathrm{d}x =
2   \left[\frac{x^3}{3}\right]_0^1 =
3   \frac{1}{3}$. \medskip
4
5 $\lim_{x \to +\infty} e^x = +\infty$. \par
6 On parle de la fonction $x \mapsto \exp(x)$. \medskip
7
8 $\lim_{x \to 0^+} \ln(x) = -\infty$ ou encore \par
9 $\lim_{\substack{x \to 0 \\ x > 0}} \ln(x) = -\infty$. \medskip
10
11
12 $a \equiv b \pmod n \Leftrightarrow \exists k \in \mathbb{Z}, a - b = kn$. \medskip
13
14
15 $\left| e^{i\theta} \right| = 1$.
16
17 ou
18
19 $\left| e^{i\theta} \right| = 1$.

```

C En licence

Finissons par une dernière définition.

Fonction continue

```

1 \textbf{Définition.} Soient $f$ une fonction définie sur une partie $A$ de $\mathbb{R}$
2 et un élément $a$ de $A$. \par
3 On dit que $f$ est \textbf{continue} au point $a$ lorsque :
4 $\forall \epsilon > 0, \exists \eta > 0, \forall x \in A, |x - a| < \eta \Rightarrow |f(x) - f(a)| < \epsilon$.

```

Ce code source donne :

Définition. Soient f une fonction définie sur une partie A de \mathbb{R} et un élément a de A .

On dit que f est **continue** au point a lorsque :

$$\forall \epsilon > 0, \exists \eta > 0, \forall x \in A, |x - a| < \eta \Rightarrow |f(x) - f(a)| < \epsilon$$

IV Aligner des égalités

Lorsque l'on veut écrire les étapes d'un long calcul, la plupart du temps, les différentes étapes sont écrites en colonne et alignées. L'environnement `align` permet de faire cela.

⚠ L'environnement `align` contient déjà le mode « mathématiques ».

Il est donc inutile d'ajouter les symboles \$, \[ou \].

$$\begin{aligned}(2x+4)^2 &= (2x)^2 + 2 \times 2x \times 4 + 4^2 \\ &= 4x^2 + 16x + 16\end{aligned}\quad (4.1)$$

```

1 \begin{align}
2   (2x+4)^2&=(2x)^2+2\times 2x\times 4+4^2 \\
3           &=4x^2+16x+16
4 \end{align}

```

On note ici l'utilisation de deux caractères spéciaux : & et \\. Le premier permet d'identifier l'endroit où se fera l'alignement. Le deuxième permet d'indiquer un changement de ligne. On retrouvera ces deux caractères dans la composition de tableau.

On peut être gêné par l'utilisation automatique de la numérotation de toutes les lignes. La commande \nonumber règle le problème. La commande \tag<texte> permet quand à elle de passer outre la numérotation automatique.

$$\begin{aligned}(2x+4)^2 &= (2x)^2 + 2 \times 2x \times 4 + 4^2 \\ (id. remarquable) \\ &= 4x^2 + 16x + 16 \quad (A) \\ &= 4x^2 + 16x + 16 \quad (354)\end{aligned}$$

```

1 \begin{align}
2   (2x+4)^2&=(2x)^2+2\times 2x\times 4
3           + 4^2\nonumber\\
4   \shortintertext{\textit{(id. remarquable)}}
5           &=4x^2+16x+16 \tag{A} \\
6           &=4x^2+16x+16 \tag{354}
7 \end{align}

```



La commande \shortintertext<texte> est ici très pratique pour insérer du texte au milieu d'une suite de calcul.

Finalement, si on ne souhaite numéroter aucune ligne, on écrira \begin{align*} et \end{align*}.

$$\begin{aligned}(2x+4)^2 &= (2x)^2 + 2 \times 2x \times 4 + 4^2 \\ &= 4x^2 + 16x + 16\end{aligned}$$

```

1 \begin{align*}
2   (2x+4)^2&=(2x)^2+2\times 2x\times 4+4^2 \\
3           &=4x^2+16x+16
4 \end{align*}

```

Remarque :

On peut également utiliser l'environnement aligned.

Celui permet également d'aligner des équations, sans les numéroter, et nécessite les balises \$...\$ ou \[...\] lorsqu'on l'utilise avec des formules mathématiques.

V Écrire un système

Pour écrire un système, il y a plusieurs possibilités que l'on retrouvera dans la section IIB du chapitre 5.

Tableaux

I Composer un tableau

A Créer des lignes et des colonnes

Pour créer un tableau, aucun *package* supplémentaire n'est chargé. L'environnement `tabular` sera utilisé en spécifiant un argument obligatoire composé d'une des trois lettres suivantes : l, c ou r. On peut inscrire plusieurs de ces lettres et une même lettre peut apparaître plusieurs fois.

Le nombre de lettres inscrites correspond simplement au nombre de colonne que contiendra le tableau. Dans chacune des colonnes, le texte pourra alors être aligné à gauche (l), centré (c) ou aligné à droite (r). Les lettres sont des *spécificateurs de colonnes*.

Ainsi, l'instruction `\begin{tabular}{rcl}` permettra la création d'un tableau contenant trois colonnes, dont le texte sera aligné à droite dans la première colonne, centré dans la deuxième et enfin aligné à gauche dans la troisième. Il est inutile de spécifier le nombre de lignes.

Article	Couleur	Prix en euros
Pantalon	bleu	25
Gants	blanc	15
	Total	40

```
1 \begin{tabular}{rcl}
2   Article & Couleur & Prix en euros \\[0.5cm]
3   Pantalon & bleu & 25 \\
4   Gants & blanc & 15 \\
5           & Total & 40
6 \end{tabular}
```

Comme pour l'environnement `align`, `&` spécifie l'emplacement de l'alignement : autrement dit ce symbole sépare chaque colonne, y compris les colonnes vides.

La commande de changement de ligne `\\` admet un argument optionnel qui sert à indiquer l'espace verticale que l'on veut insérer après cette ligne.

Afin de mieux visualiser les différentes lignes et colonnes, on peut ajouter des *filets* : la commande `\hline` trace un *filet horizontal*. Pour insérer un *filet vertical*, on peut inscrire une barre verticale `|` entre les deux spécificateurs de colonnes concernés.

Article	Couleur	Prix en euros
Pantalon	bleu	25
Gants	blanc	15
	Total	40

```
1 \begin{tabular}{r|c|l}
2   Article & Couleur & Prix en euros \\ \hline \hline
3   Pantalon & bleu & 25 \\
4   Gants & blanc & 15 \\ \hline
5           & Total & 40
6 \end{tabular}
```

B Fusionner des colonnes

TEX offre la commande `\multicolumn<nombre-colonnes><specificateur><texte>` qui permet de fusionner horizontalement des colonnes.

Article	Couleur	Prix en euros
Pantalon	bleu	25
Gants	blanc	15
Total :		40

```
1 \begin{tabular}{r|c|l}
2   \multicolumn{1}{c|}{Article} &
3   Couleur & Prix en euros \\ \hline
4   Pantalon & bleu & 25 \\
5   Gants & blanc & 15 \\ \hline
6   \multicolumn{2}{r}{Total :} & 40
7 \end{tabular}
```



On constate dans l'exemple précédent que la commande `\multicolumn` permet non seulement de fusionner plusieurs colonnes mais aussi de modifier ponctuellement le spécificateur de colonne sur une cellule en particulier.

C Spécificateurs de colonne supplémentaires

Les trois spécificateurs de base ne permettent pas de changement de ligne au sein d'une même cellule. De plus, on ne peut pas spécifier la largeur de la cellule. Le spécificateur `p{<dim>}` permet de composer une colonne en imposant une largeur : le texte est alors composé comme un paragraphe normal.

Rectangle	Quadrilatère dont les diagonales sont de même longueur et se coupent en leur milieu. Le rectangle est un parallélogramme.
-----------	--

```
1 \begin{tabular}{r|p{4.5cm}}
2   Rectangle & Quadrilatère dont les diagonales
3   sont de même longueur et
4   se coupent en leur milieu.\par
5   Le rectangle est un parallélogramme.
6 \end{tabular}
```

Dans l'exemple précédent, on constate que les deux paragraphes sont alignés sur le haut de la cellule. On pourrait avoir envie de centrer verticalement ces paragraphes. On utilisera simplement `m{<dim>}` :

Rectangle	Quadrilatère dont les diagonales sont de même longueur et se coupent en leur milieu. Le rectangle est un parallélogramme.
-----------	--

```
1 \begin{tabular}{r|m{4.5cm}}
2   Rectangle & Quadrilatère dont les diagonales
3   sont de même longueur et
4   se coupent en leur milieu.\par
5   Le rectangle est un parallélogramme.
6 \end{tabular}
```



Dans l'exemple précédent, si l'on remplace `m{4.5cm}` par `b{4.5cm}` le texte sera aligné en bas verticalement.

II Les tableaux en mathématiques

A

L'environnement `array`

Comme nous allons rapidement le voir, il peut être utile de composer des tableaux en mode mathématique (notamment en mode *hors texte*). Dans ce cas, on utilise dans le mode l'environnement `array` qui fonctionne comme `tabular` pour les commandes expliquées dans la section précédente.

$$\begin{array}{lcl} f: \mathbb{R} & \rightarrow & \mathbb{R}_+ \\ x & \mapsto & x^2 \end{array}$$

```
1 \[\begin{array}{rcl}
2   f \colon \mathbb{R} & \rightarrow & \mathbb{R}_+ \\
3   x & \mapsto & x^2
4 \end{array}\]
```

B

Les systèmes

On avait constaté l'utilité des symboles étirables horizontalement. Servons-nous de `\left\{` pour créer une grande accolade pour les systèmes :

$$\left\{ \begin{array}{lcl} 2x & - & y & - & 3z & = & 1 \\ 3x & + & 2y & & & = & -4 \\ -x & & & + & 6z & = & 22 \end{array} \right.$$

```
1 \[\left\{
2   \begin{array}{cccccl}
3     2x & - & y & - & 3z & = & 1 \\
4     3x & + & 2y & & & = & -4 \\
5     -x & & & + & 6z & = & 22
6   \end{array}
7 \right.\]
```



`\left` et `\right` fonctionne toujours ensemble. Ici, on ne veut qu'une accolade à gauche. On utilise donc `\left\{` mais la commande `\right.` est nécessaire pour que la compilation ait lieu sans erreur.

$$|x| = \begin{cases} -x & \text{si } x < 0 \\ x & \text{sinon} \end{cases}$$

```
1 \[\lvert x \rvert =
2 \left\{
3   \begin{array}{cl}
4     -x & \text{si } x < 0 \\
5     x & \text{sinon}
6   \end{array}
7 \right.\]
```

Il existe un environnement permettant d'écrire des systèmes de façon rapide. Il s'agit de l'environnement *cases*. Il nécessite le *package* *amsmath* et aligne les équations à gauche.

$$\begin{cases} 2x + 4y - 5z + 1 = 0 \\ x - y - 5 = 0 \end{cases}$$

```

1 \[ \begin{cases}
2 2x+4y-5z+1=0 \\
3 x-y-5=0
4 \end{cases} \]
```

Les matrices

Cette fois-ci, on peut penser à utiliser `\left(` et `\right)`...

$$M = \begin{pmatrix} 2 & -1 & 0 \\ 3 & 4 & 1 \\ 0 & 2 & 3 \end{pmatrix}$$

```

1 \[ M = \left(
2   \begin{array}{ccc}
3     2 & -1 & 0 \\
4     3 & 4 & 1 \\
5     0 & 2 & 3
6   \end{array}
7 \right) \]
```

... mais en réalité des environnements spécifiques existent pour l'écriture des matrices. Entre autre, on retiendra les environnements `pmatrix` et `vmatrix` ainsi que la commande `\bordermatrix`.

$$M = \begin{pmatrix} 2 & -1 & 0 \\ 3 & 4 & 1 \\ 0 & 2 & 3 \end{pmatrix}$$

$$\det(M) = \begin{vmatrix} 2 & -1 & 0 \\ 3 & 4 & 1 \\ 0 & 2 & 3 \end{vmatrix} = 29$$

$$\begin{matrix} & f(e_1) & f(e_2) \\ e_1 & \begin{pmatrix} 1 & 2 \end{pmatrix} \\ e_2 & \begin{pmatrix} 0 & 3 \end{pmatrix} \end{matrix}$$

```

1 \[ M =
2   \begin{pmatrix}
3     2 & -1 & 0 \\
4     3 & 4 & 1 \\
5     0 & 2 & 3
6   \end{pmatrix}
7 \] \medskip
8 \[ \det(M) =
9   \begin{vmatrix}
10    2 & -1 & 0 \\
11    3 & 4 & 1 \\
12    0 & 2 & 3
13   \end{vmatrix} = 29
14 \] \medskip
15 \[
16   \bordermatrix{
17     & f(e_1) & f(e_2) \\
18   e_1 & 1 & 2 \\
19   e_2 & 0 & 3
20 }
```

III Tableaux de signes et tableaux de variations

Ici du allons utiliser l'environnement `tikzpicture` du *package* TikZ. Nous ne dirons rien sur ce *package* pour le moment mais nous reviendrons dessus lorsque nous traiterons les graphiques. Cependant, pour créer des tableaux de signes et des tableaux de valeurs, Alain MATTHES a créé le *package* `tkz-tab` qui répond parfaitement bien à notre problème. Il faut donc penser dès à présent à ajouter `\usepackage{tkz-tab}` au préambule.



Selon l'installation effectuée, il est possible que `tkz-tab` ne soit pas présent dans votre distribution. Vous vous en rendrez rapidement compte en compilant le premier exemple. La documentation de ce *package* est également une source à consulter absolument.

A Tableaux de signes

```

1 \textbf{Tableau de signes :}\par
2 \begin{tikzpicture}
3   \tkzTabInit[nocadre,espcl=1.5,lgt=2.5]%
4     {\$x$/0.75,Signe de \$\$ \$x+2\$/1.5,Signe de \$\$ \$x^2-1\$/1.5,%
5      signe du \$\$ produit/1.5,signe du \$\$ quotient/1.5}%
6     {\$-\infty\$, \$-2\$, \$-1\$, \$1\$, \$+\infty\$}
7   \tkzTabLine{,-,z,+,t,+,t,+}
8   \tkzTabLine{+,t,+,z,-,z,+}
9   \tkzTabLine{,-,z,+,z,-,z,+}
10  \tkzTabLine{,-,z,+,d,-,d,+}
11 \end{tikzpicture}

```

Tableau de signes :						
x	$-\infty$	-2	-1	1	$+\infty$	
Signe de $x + 2$	-	0	+	+	+	
Signe de $x^2 - 1$	+	+	0	-	0	+
signe du produit	-	0	+	0	0	+
signe du quotient	-	0	+	-	+	

Un tableau de signes ou de variations commencera toujours par `\tkzTabInit` dont voilà la syntaxe :

`\tkzTabInit[<options>]{<première colonne>}{<première ligne>}`

En option est spécifié que le cadre autour du tableau ne doit pas être dessiné, que l'espace entre les valeurs de la première ligne est réglée par le paramètre `espc1` et que la largeur de la première colonne est réglée par le paramètre `lgt`.

Par défaut les valeurs de `espc1` et `lgt` sont réglées à 2 cm.

Il existe également d'autres options :

- ★ `deltac1` : marge avant le premier antécédent et après le dernier, réglée à 0,5 cm par défaut,
- ★ `lw` : épaisseur des lignes du tableau, réglée à 0,4 pt par défaut,
- ★ `nocadre` : pour enlever le cadre entourant le tableau ; par défaut, on encadre le tableau
- ★ `color` : booléen qui autorise la couleur
- ★ `colorC` : couleur de la première colonne, réglée à `white` par défaut
- ★ `colorL` : couleur de la première ligne, réglée à `white` par défaut
- ★ `colorT` : couleur de la partie centrale, réglée à `white` par défaut
- ★ `colorV` : couleur de la case de la variable, réglée à `white` par défaut

Ensuite, on énumère les différentes lignes de la première colonne : nom de la ligne/hauteur de la ligne. Le nom de la ligne accepte des changements de lignes à l'aide de `\\` et on passe d'une ligne à l'autre en utilisant la virgule.

Enfin, dans le dernier argument, on écrit les valeurs de la première ligne en les séparant par une virgule.

Pour finir, pour chaque ligne, on écrit les signes par la commande `\tkzTabLine`. La lettre `t` crée un filet vertical en pointillés et la lettre `z` fait la même chose en ajoutant un zéro. La lettre `d` insère une double-barre pour les valeurs interdites.



Si on a besoin d'écrire des nombres décimaux, on prendra soin de les écrire entre accolades pour que la virgule ne crée pas de conflit : `$(1,5)$`.

B

Tableaux de variations

```
1 \textbf{Tableau de variations :}\par
2 \begin{tikzpicture}
3   \tkzTabInit[nocadre,espc1=2,lgt=2.5]
4     {$x$/0.75,Variations \\ de $f$/1.75}
5     {$-3$, $-1$, $1$, $4$}
6     \tkzTabVar{+/$6$, -D+/$-\infty$/$+\infty$, -/, +/$-5$}
7 \end{tikzpicture}
```

Tableau de variations :

x	-3	-1	1	4
Variations de f	6 ↘ $-\infty$	$+\infty$ ↘ ↗ -5		

Il nous suffit ici de commenter la commande `\tkzTabVar` : pour chaque valeurs de x indiquée sur la première ligne, on peut préciser une valeur précédée du signe $+/$ pour dire que la valeur sera écrite « en haut » ou bien $-/$ pour dire que la valeur sera écrite « en bas ». Des flèches relieront alors automatiquement les différentes valeurs :

`\tkzTabVar{<+ ou ->/<valeur>,<+ ou ->/<valeur>,...}`



Pour la double-barre avec des valeurs indiquées avant et après celle-ci, on notera `-D+<valeur>/<valeur>` ou bien `+D-<valeur>/<valeur>`.

C Un mélange

Et voilà maintenant ce que l'on peut obtenir :

```

1 \begin{tikzpicture}
2   \tkzTabInit[nocadre,espc1=2,lgt=2.5]%
3     {\$x\$/0.75,signe de \$\$ f'(x)\$/1.5,Variations \$\$ de \$f\$/1.75}%
4     {\$-3\$, \$-1\$, \$1\$, \$4\$}
5   \tkzTabLine{+,z,-,z,+}
6   \tkzTabVar{-/\$-\infty\$,+/\$-/,+/\$+\infty\$}
7 \end{tikzpicture}

```

x	-3	-1	1	4	
signe de $f'(x)$	+	0	-	0	+
Variations de f	$-\infty$				$+\infty$

D Un dernier exemple

Avec l'instruction `\tkzTabIma` qui permet des images intermédiaires.

```

1. \begin{tikzpicture}
2. \tkzTabInit[color,lgt=2.5,espcl=3,colorC=blue!20,colorV=blue!20]%
3. {\$t$/1,Variations\\ de \$f$/2,Variations\\ de \$g$/2}%
4. {\$-1$, \$0$, \$1$, \$2$}
5. \tkzTabVar{-/-$-5$,+/$0$, -/-$-1$,+/$4$}
6. \tkzTabVar{-/-$-5$,R/,R/,+/$4$}
7. \tkzTabIma[draw]{1}{4}{2}{\$0$}
8. \tkzTabIma[draw]{1}{4}{3}{\$3$}
9. \end{tikzpicture}

```

t	-1	0	1	2
Variations de f		0	-1	4
Variations de g		0	3	4

⚠ Pour placer une image entre deux autres, il faut que les deux images extrêmes existent...
Il ne faut donc pas utiliser une image qui a été remplacée par R...

IV Complément : le tableur

Le *package* pas-tableur de Stéphane PASQUET permet d'imiter l'apparence d'un tableur. Cependant, il n'effectue pas de calcul automatique comme dans un tableur (pour cela, on pourra jeter un œil sur le *package* spreadtab). Une fois encore, ce *package* utilise TikZ et son environnement tikzpicture.

La première commande à retenir est la suivante :

```
\tableur[<nombres-lignes>]{<colonnes>}
```

L'argument <colonnes> permet de spécifier les lettres des colonnes utilisées soit une par une, soit en utilisant un « intervalle ».

Ensuite,

```
\celtxt[<options>]{<colonne>}{<ligne>}{<texte>}
```

permet d'écrire un texte dans la cellule définie par <colonne> et <ligne>. Parmi les options, l, c et r sont utilisées pour l'alignement du texte. Des formules commençant par le signe = peuvent être écrites et le texte peut être mis en forme en utilisant les commandes correspondantes.

Et enfin, les commandes \selecCell et \multiSelec permettent de mettre en couleur des cellules sélectionnées. Les exemples suivants vous montrent comment :

	A	D	T
1			

	A	B	C
1			
2			
3			

	A	B	C	D
1	x	$f(x)$		
2	0	$=A2*A2$		
3	1			

```

1 \begin{tikzpicture}
2   \tabcolwidth{1cm}
3   \tableur{A,D,T}
4 \end{tikzpicture}\bigskip
5
6 \begin{tikzpicture}
7   \tabcolwidth{2cm}
8   \tableur[3]{A,B,C}
9   \multiSelec{A-2}{B-3}
10 \end{tikzpicture}\bigskip
11
12 \begin{tikzpicture}
13   \tabcolwidth{1.5cm}
14   \tableur[3]{A-D}
15   \celtxt[c]{A}{1}{\itshape x}
16   \celtxt[r]{A}{2}{0}
17   \celtxt[r]{A}{3}{1}
18   \celtxt[c]{B}{1}{\itshape f(x)}
19   \celtxt{B}{2}{=A2*A2}
20   \selecCell{B}{2}
21 \end{tikzpicture}

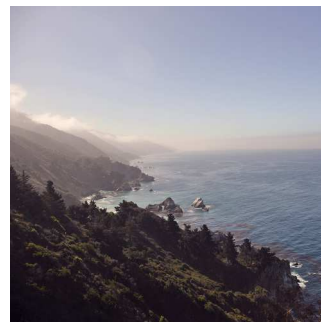
```

I Insérer une image

Pour insérer une image, un package intéressant est le package *graphicx* (Ajouter la commande `\usepackage{graphicx}` dans le préambule).

Il fournit la commande `\includegraphics` qui possède des options facultatives.

```
\includegraphics{exemples/image}
```



Les formats des images acceptés par compilation PDF \LaTeX sont *.jpg*, *.png* et *.pdf*.

Le format accepté par compilation \LaTeX est *.eps*.

Il est inutile de préciser l'extension du fichier utilisé. Il faut juste utiliser un fichier dont le format est le bon, suivant la compilation choisie.



Si l'on veut centrer une image, il suffit d'utiliser l'environnement *center* comme pour le texte.

II Modifier une image

La commande plusieurs `\includegraphics` admet plusieurs options que l'on peut spécifier entre crochets. Ces options permettent de transformer l'aspect de l'image.

Parmi les options possibles, on trouve :

★ *width* : par exemple `width=1cm` fixe la largeur de l'image

On peut également utiliser `width=\linewidth` ou `width=\textwidth`, `width=0.5\linewidth`

★ *height* : par exemple `height=2cm` fixe la hauteur de l'image

★ *scale* : par exemple `scale=0.5` met l'image à l'échelle 0.5

★ *angle* : par exemple `angle=45` effectue une rotation de l'image de 45° dans le sens positif, autour du centre de l'image.

```
\includegraphics[angle=45,width=0.6\linewidth]{exemples/image}
```



III Les environnements *wrapfigure* et *minipage*

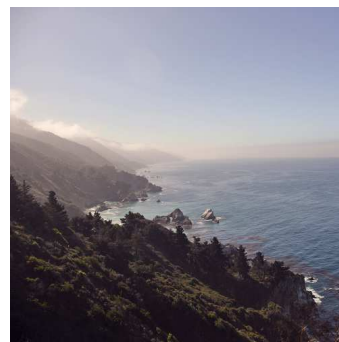
Pour utiliser l'environnement *wrapfigure*, il faut charger, dans le préambule, le package *wrapfigure*.

Environnement *wrapfigure*

```
1 \begin{wrapfigure}{r}{5cm}
2 \includegraphics[width=4.5cm]{exemples/image}
3 \end{wrapfigure}
4 \LaTeX{}, c'est génial ! \LaTeX{}, c'est génial ! \LaTeX{}, c'est génial !
5 \LaTeX{}, c'est génial ! \LaTeX{}, c'est génial ! \LaTeX{}, c'est génial !
6 \LaTeX{}, c'est génial ! \LaTeX{}, c'est génial ! \LaTeX{}, c'est génial !
7 \LaTeX{}, c'est génial ! \LaTeX{}, c'est génial ! \LaTeX{}, c'est génial !
8 \LaTeX{}, c'est génial ! \LaTeX{}, c'est génial ! \LaTeX{}, c'est génial !
9 \LaTeX{}, c'est génial ! \LaTeX{}, c'est génial ! \LaTeX{}, c'est génial !
10 \LaTeX{}, c'est génial ! \LaTeX{}, c'est génial ! \LaTeX{}, c'est génial !
11 \LaTeX{}, c'est génial ! \LaTeX{}, c'est génial ! \LaTeX{}, c'est génial !
12 \LaTeX{}, c'est génial ! \LaTeX{}, c'est génial ! \LaTeX{}, c'est génial !
13 \LaTeX{}, c'est génial ! \LaTeX{}, c'est génial ! \LaTeX{}, c'est génial !
14 \LaTeX{}, c'est génial ! \LaTeX{}, c'est génial ! \LaTeX{}, c'est génial !
15 \LaTeX{}, c'est génial ! \LaTeX{}, c'est génial ! \LaTeX{}, c'est génial !
16 \LaTeX{}, c'est génial ! \LaTeX{}, c'est génial ! \LaTeX{}, c'est génial !
17 \LaTeX{}, c'est génial ! \LaTeX{}, c'est génial ! \LaTeX{}, c'est génial !
```

Ce code source donne :

\LaTeX , c'est génial ! \LaTeX , c'est génial ! \LaTeX , c'est génial ! \LaTeX , c'est
génial ! \LaTeX , c'est génial ! \LaTeX , c'est génial ! \LaTeX , c'est génial ! \LaTeX ,
c'est génial ! \LaTeX , c'est génial ! \LaTeX , c'est génial ! \LaTeX , c'est génial !
 \LaTeX , c'est génial ! \LaTeX , c'est génial ! \LaTeX , c'est génial ! \LaTeX , c'est
génial ! \LaTeX , c'est génial ! \LaTeX , c'est génial ! \LaTeX , c'est génial ! \LaTeX ,
c'est génial ! \LaTeX , c'est génial ! \LaTeX , c'est génial ! \LaTeX , c'est génial !
 \LaTeX , c'est génial ! \LaTeX , c'est génial ! \LaTeX , c'est génial ! \LaTeX , c'est
génial ! \LaTeX , c'est génial ! \LaTeX , c'est génial ! \LaTeX , c'est génial ! \LaTeX ,
c'est génial ! \LaTeX , c'est génial ! \LaTeX , c'est génial ! \LaTeX , c'est génial !
 \LaTeX , c'est génial ! \LaTeX , c'est génial ! \LaTeX , c'est génial ! \LaTeX , c'est
génial ! \LaTeX , c'est génial ! \LaTeX , c'est génial ! \LaTeX , c'est génial ! \LaTeX , c'est génial !



L'image peut être remplacée par une figure (par exemple *tikzpicture*),
du texte, ...

Personnellement, j'utilise plutôt l'environnement *minipage* qui permet de horizontalement découper la page en plusieurs « sous-pages ».

minipage s'utilise ainsi :

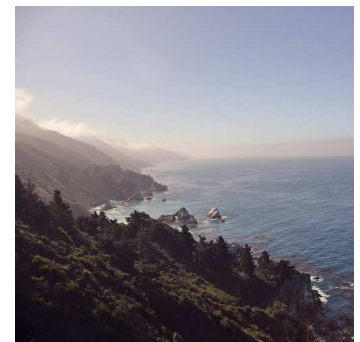
```

1 \begin{minipage}{0.7\linewidth}
2 \LaTeX{}, c'est génial ! \LaTeX{}, c'est génial ! \LaTeX{}, c'est génial !
3 \LaTeX{}, c'est génial ! \LaTeX{}, c'est génial ! \LaTeX{}, c'est génial !
4 \LaTeX{}, c'est génial ! \LaTeX{}, c'est génial ! \LaTeX{}, c'est génial !
5 \LaTeX{}, c'est génial ! \LaTeX{}, c'est génial ! \LaTeX{}, c'est génial !
6 \LaTeX{}, c'est génial ! \LaTeX{}, c'est génial ! \LaTeX{}, c'est génial !
7 \LaTeX{}, c'est génial ! \LaTeX{}, c'est génial ! \LaTeX{}, c'est génial !
8 \LaTeX{}, c'est génial ! \LaTeX{}, c'est génial ! \LaTeX{}, c'est génial !
9 \LaTeX{}, c'est génial ! \LaTeX{}, c'est génial ! \LaTeX{}, c'est génial !
10 \LaTeX{}, c'est génial ! \LaTeX{}, c'est génial ! \LaTeX{}, c'est génial !
11 \LaTeX{}, c'est génial ! \LaTeX{}, c'est génial ! \LaTeX{}, c'est génial !
12 \LaTeX{}, c'est génial ! \LaTeX{}, c'est génial ! \LaTeX{}, c'est génial !
13 \LaTeX{}, c'est génial ! \LaTeX{}, c'est génial ! \LaTeX{}, c'est génial !
14 \end{minipage}
15 \hfill
16 \begin{minipage}{0.25\linewidth}
17 \includegraphics[width=4.5cm]{exemples/image}
18 \end{minipage}
19
20 \medskip
21
22 \LaTeX{}, c'est génial ! \LaTeX{}, c'est génial ! \LaTeX{}, c'est génial !
23 \LaTeX{}, c'est génial ! \LaTeX{}, c'est génial ! \LaTeX{}, c'est génial !
24 \LaTeX{}, c'est génial ! \LaTeX{}, c'est génial ! \LaTeX{}, c'est génial !

```

Ce *code source* donne :

\LaTeX , c'est génial! \LaTeX , c'est génial! \LaTeX , c'est génial! \LaTeX , c'est
 génial! \LaTeX , c'est génial! \LaTeX , c'est génial! \LaTeX , c'est génial! \LaTeX ,
 c'est génial! \LaTeX , c'est génial! \LaTeX , c'est génial! \LaTeX , c'est génial!
 \LaTeX , c'est génial! \LaTeX , c'est génial! \LaTeX , c'est génial! \LaTeX , c'est
 génial! \LaTeX , c'est génial! \LaTeX , c'est génial! \LaTeX , c'est génial! \LaTeX ,
 c'est génial! \LaTeX , c'est génial! \LaTeX , c'est génial! \LaTeX , c'est génial!
 \LaTeX , c'est génial! \LaTeX , c'est génial! \LaTeX , c'est génial! \LaTeX , c'est
 génial! \LaTeX , c'est génial! \LaTeX , c'est génial! \LaTeX , c'est génial!
 \LaTeX , c'est génial! \LaTeX , c'est génial! \LaTeX , c'est génial!
 \LaTeX , c'est génial! \LaTeX , c'est génial! \LaTeX , c'est génial! \LaTeX , c'est génial! \LaTeX ,
 c'est génial! \LaTeX , c'est génial! \LaTeX , c'est génial! \LaTeX , c'est génial!



IV Une image dans du texte



On désire parfois insérer une image dans une ligne de texte.

On peut, par exemple, utiliser la commande `\raisebox` pour cela.

Commande `raisebox`

```
1 Voici du texte et une image à l'intérieur
2 sans raisebox
3 \includegraphics[scale=0.3]{exemples/image}
4 \dots et avec raisebox \raisebox{-1ex}{
5 \includegraphics[scale=0.3]{exemples/image}}.
6
7 Ce n'est pas toujours heureux
8 typographiquement parlant.
```

Ce *code source* donne :

Voici du texte et une image à l'intérieur sans `raisebox`  ... et avec `raisebox` .
Ce n'est pas toujours heureux typographiquement parlant.



La commande `\raisebox` n'est pas réservée aux images.

Commande `raisebox`

```
1 Du texte \raisebox{-1ex}{en bas}
2 et du texte \raisebox{1ex}{en haut}.
```

Ce *code source* donne :

Du texte_{en bas} et du texte^{en haut}.

Graphiques et géométrie

I Différentes méthodes

★ Le package *pstricks* a été longtemps très largement utilisé pour réaliser des figures géométriques et graphiques.

Il l'est toujours actuellement et offre de nombreuses possibilités pour créer des graphiques et figures géométriques.

Cependant, on rencontre des problèmes de compatibilité lorsque l'on compile en PDF \LaTeX .

★ Le programme *metapost* est également très puissant et offre de nombreuses possibilités.

Cependant il génère des fichiers externes qu'il faut ensuite inclure au document.

★ Mon choix s'est donc porté sur le package *TikZ* qui possède un très grand nombre d'options et de possibilités, ainsi que sur un dérivé, le package *TkZ* qui utilise le premier et offre une syntaxe simplifiée dans pas mal de cas de figures.

II Présentation générale de TikZ et TkZ

Les packages *TikZ* et *TkZ* possèdent des bibliothèques.

Pour les utiliser il faut ajouter dans le préambule :

```
\usepackage{tikz,tkz-base,tkz-fct,tkz-euclide,tkz-tab,tkz-graph,tikz-3dplot}
```

```
\usetikzlibrary{calc,shapes,arrows,plotmarks,lindenmayersystems,decorations,  
decorations.markings,decorations.pathmorphing,  
decorations.pathreplacing,patterns,positioning,decorations.text}
```

Pour utiliser les différents objets *TkZ*, on ajoute :

```
\usetkzobj{all}
```

Certaines commandes *TkZ* nécessitent d'effectuer des calculs pour obtenir les figures et graphiques et utilisent le package *pgfplots*.

On ajoute donc également la commande `\usepackage{pgfplots}`

On ajoutera enfin l'appel au package *etex* qui augmente la capacité de mémoire de \LaTeX , ce qui peut s'avérer nécessaire lorsque le document contient pas mal de figures nécessitant beaucoup de calculs.

Voici donc un préambule possible tenant compte de tous ces *packages* :

Document TikZ

```
1 \documentclass[12pt,french]{article}  
2 \usepackage{etex}  
3 \usepackage[utf8x]{inputenc}  
4 \usepackage[T1]{fontenc}
```

```

5 \usepackage[upright]{fourier}
6 \usepackage[a4paper,margin=2cm]{geometry}
7 \usepackage{mathtools,amssymb}
8 \usepackage{babel}
9 \usepackage{tikz,tkz-base,tkz-fct,tkz-euclide,tkz-tab,tkz-graph,tkz-3dplot}
10 \usetikzlibrary{calc,shapes,arrows,plotmarks,lindenmayersystems,decorations,
11 decorations.markings,decorations.pathmorphing,
12 decorations.pathreplacing,patterns,positioning,decorations.text}
13 \usetikzobj{all}
14 \usepackage{pgfplots}
15
16 \begin{document}
17
18 \begin{tikzpicture}
19 < ...instructions... >
20 \end{tikzpicture}
21
22 \end{document}
23

```

III Exemples de figures utilisant TikZ



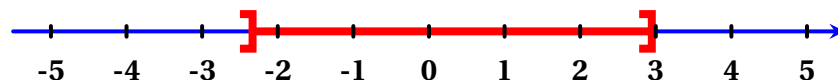
Ces figures sont réalisables avec des commandes des *packages* TkZ que l'on voit juste après.

Axe gradué

```

1 \begin{center}
2 \begin{tikzpicture}[scale=1,line cap=round,line join=round,x=1.0cm]
3 \draw[>=stealth,->,line width=1.5pt,color=blue] (-5.5,0) -- (5.5,0);
4 \draw[{}-{}],line width=3pt,color=red] (-2.5,0) -- (3,0);
5 \foreach \x in {-5,...,5}
6 \draw[color=black,line width=1.5pt] (\x,2pt) -- (\x,-2pt);
7 \foreach \x in {-5,...,5}
8 \draw (\x,-7pt) node[below,font=\bfseries] {\x};
9 \end{tikzpicture}
10 \end{center}

```

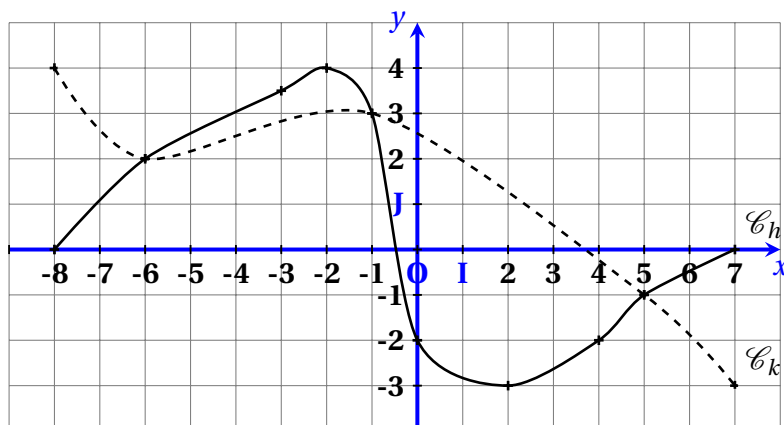


—— Courbe passant par des points donnés ——

```

1 \begin{center}
2 \begin{tikzpicture}[scale=0.6,>=stealth]
3     \draw[gray] (-9,-4)grid(8,5);
4     \draw[->,line width=1.5pt,color=blue] (-9,0)--(8,0)node[below] {$x$};
5     \foreach \x in {-8,-7,-6,-5,-4,-3,-2,-1,2,3,4,5,6,7}
6 \draw[color=black] (\x,2pt) -- (\x,-2pt) node[below,font=\bfseries] {\x};
7     \draw[->,line width=1.5pt,color=blue] (0,-4)--(0,5)node[left] {$y$};
8     \foreach \y in {-3,-2,-1,2,3,4}
9 \draw[color=black] (2pt,\y) -- (-2pt,\y) node[left,font=\bfseries] {\y};
10    \coordinate (O) at (0,-2pt); \draw (O) node[below,font=\bfseries,color=blue] {0};
11    \coordinate (I) at (1,-2pt); \draw (I) node[below,font=\bfseries,color=blue] {I};
12    \foreach \x in {-9,...,7}
13    \draw[line width=0.7pt] (\x,-0.1)--(\x,0.1);
14    \coordinate (J) at (-2pt,1); \draw (J) node[left,font=\bfseries,color=blue] {J};
15    \foreach \x in {-3,...,4}
16    \draw[line width=0.7pt] (-0.1,\x)--(0.1,\x);
17    \draw[line width=1pt] plot[smooth=200,mark=+,mark options={scale =1.5}]
18    coordinates{(-8,0)(-6,2)(-3,3.5)(-2,4)(-1,3)(0,-2)(2,-3)(4,-2)(5,-1)(7,0)}
19    node[above right]{$\mathcal{C}_h$};
20    \draw[dashed, line width=1pt] plot[smooth=200,mark=+,mark options={scale =1.5}]
21    coordinates{(-8,4)(-6,2)(-1,3)(5,-1)(7,-3)} node[above right]{$\mathcal{C}_k$};
22 \end{tikzpicture}
23 \end{center}

```



—— Diagrammes de Venn ——

```

1 \begin{center}
2 \def\firstcircle{(0,0) circle (1.5cm)}
3 \def\secondcircle{(45:2cm) circle (1.5cm)}
4 \def\thirdcircle{(0:2cm) circle (1.5cm)}
5
6 \begin{tikzpicture}
7     \draw \firstcircle node[below] {$A$};
8     \draw \secondcircle node [above] {$B$};

```

```

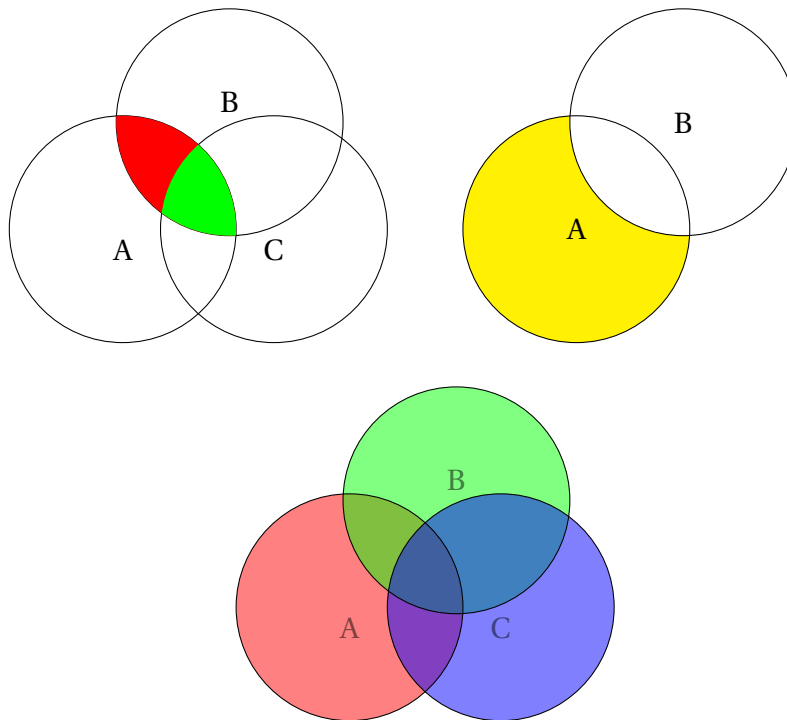
9      \draw \thirdcircle node [below] {$C$};
10
11      % Now we want to highlight the intersection of the first and the
12      % second circle:
13
14      \begin{scope}
15          \clip \firstcircle;
16          \fill[red] \secondcircle;
17      \end{scope}
18
19      % Next, we want the highlight the intersection of all three circles:
20
21      \begin{scope}
22          \clip \firstcircle;
23          \clip \secondcircle;
24          \fill[green] \thirdcircle;
25      \end{scope}
26
27      % The intersection trick works pretty well for intersections. If you need
28      % the set-theoretic difference between two sets, things are a little more
29      % complicated:
30
31      % Suppose we want to highlight the part of the first circle that is not
32      % also part of the second circle. For this, we need to clip against the
33      % "complement" of the second circle. The trick is to add a large rectangle
34      % that encompasses everything and then use the even-odd filling rule
35      % (see the manual again):
36
37      \begin{scope}[shift={(6cm,0cm)}]
38          \begin{scope}[even odd rule]% first circle without the second
39              \clip \secondcircle (-3,-3) rectangle (3,3);
40              \fill[yellow] \firstcircle;
41          \end{scope}
42          \draw \firstcircle node {$A$};
43          \draw \secondcircle node {$B$};
44      \end{scope}
45
46      % When using the above, you will notice that the border lines of the
47      % original circles are erased by the intersection parts. To solve this
48      % problem, either use a background layer (see the manual) or simply draw
49      % the border lines after everything else has been drawn.
50
51      % The last trick is to cheat and use transparency
52      \begin{scope}[shift={(3cm,-5cm)}, fill opacity=0.5]
53          \fill[red] \firstcircle;
54          \fill[green] \secondcircle;
55          \fill[blue] \thirdcircle;
56          \draw \firstcircle node[below] {$A$};
57          \draw \secondcircle node [above] {$B$};

```

```

58     \draw \thirdcircle node [below] {$C$};
59     \end{scope}
60
61
62 \end{tikzpicture}
63 \end{center}

```



Un arbre pondéré généré à partir du site <http://math.et.info.free.fr/>

Arbre pondéré

```

1 %:-+--+ Engendré par : http://math.et.info.free.fr/TikZ/Arbre/
2 \begin{center}
3 % Racine à Gauche, développement vers la droite
4 \begin{tikzpicture}[xscale=0.9,yscale=0.9]
5 % Styles (MODIFIABLES)
6 \tikzstyle{fleche}=[->,>=latex,thick,color=blue]
7 \tikzstyle{score}=[->,>=latex,thick,style=dotted,color=red]
8 \tikzstyle{noeud}=[fill=white],circle,draw]
9 \tikzstyle{feuille}=[fill=white],circle,draw]
10 \tikzstyle{feuillescore}=[fill=white,text=red],circle,draw]
11 \tikzstyle{etiquette}=[midway,fill=white],draw]
12 % Dimensions (MODIFIABLES)
13 \def\DistanceInterNiveaux{3}
14 \def\DistanceInterFeuilles{1.5}
15 % Dimensions calculées (NON MODIFIABLES)
16 \def\NiveauA{(0)*\DistanceInterNiveaux}
17 \def\NiveauB{(1.6666666666666665)*\DistanceInterNiveaux}
18 \def\NiveauC{(3)*\DistanceInterNiveaux}

```

```

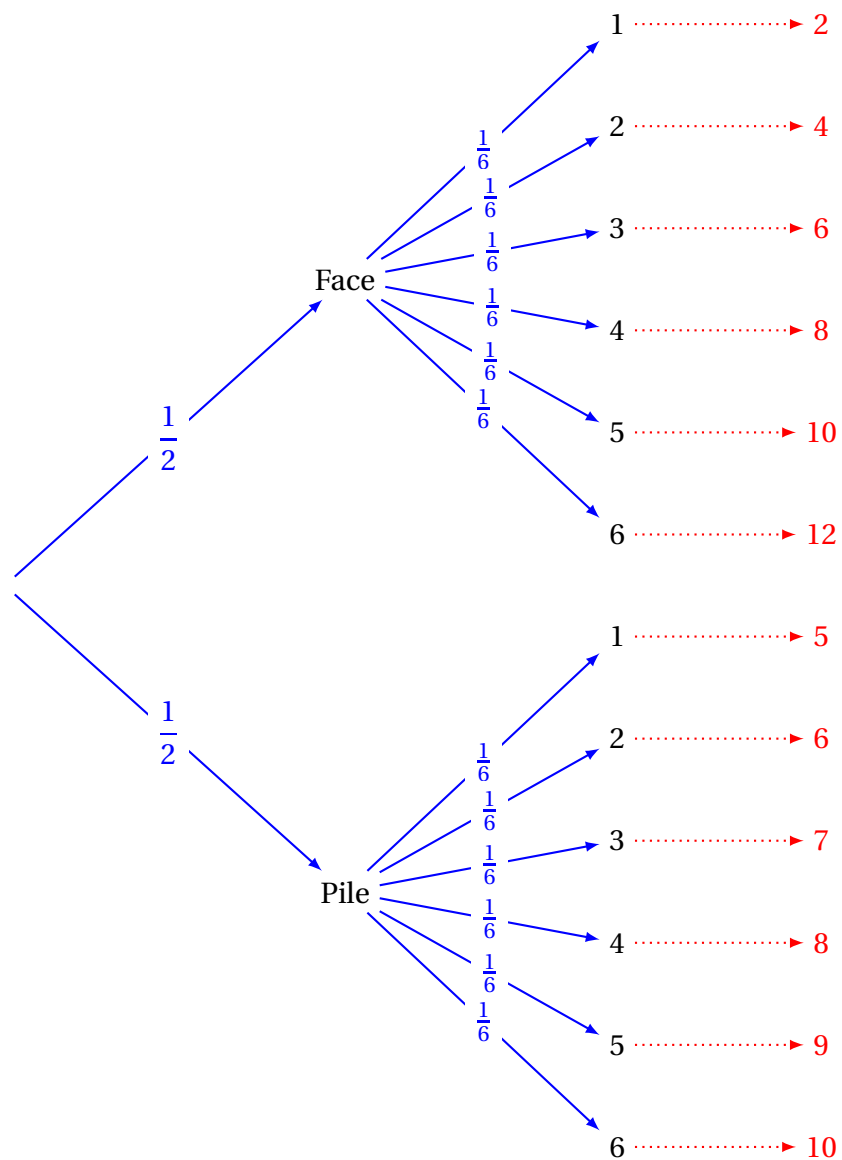
19 \def\NiveauD{(4)*\DistanceInterNiveaux}
20 \def\InterFeuilles{(-1)*\DistanceInterFeuilles}
21 % Noeuds (MODIFIABLES : Styles et Coefficients d'InterFeuilles)
22 \node[noeud] (R) at ({\NiveauA},{(5.5)*\InterFeuilles}) {};
23 \node[noeud] (Ra) at ({\NiveauB},{(2.5)*\InterFeuilles}) {Face};
24 \node[noeud] (Raa) at ({\NiveauC},{(0)*\InterFeuilles}) {$1$};
25 \node[feuillescore] (Raaa) at ({\NiveauD},{(0)*\InterFeuilles}) {$2$};
26 \node[noeud] (Rab) at ({\NiveauC},{(1)*\InterFeuilles}) {$2$};
27 \node[feuillescore] (Raba) at ({\NiveauD},{(1)*\InterFeuilles}) {$4$};
28 \node[noeud] (Rac) at ({\NiveauC},{(2)*\InterFeuilles}) {$3$};
29 \node[feuillescore] (Raca) at ({\NiveauD},{(2)*\InterFeuilles}) {$6$};
30 \node[noeud] (Rad) at ({\NiveauC},{(3)*\InterFeuilles}) {$4$};
31 \node[feuillescore] (Rada) at ({\NiveauD},{(3)*\InterFeuilles}) {$8$};
32 \node[noeud] (Rae) at ({\NiveauC},{(4)*\InterFeuilles}) {$5$};
33 \node[feuillescore] (Raea) at ({\NiveauD},{(4)*\InterFeuilles}) {$10$};
34 \node[noeud] (Raf) at ({\NiveauC},{(5)*\InterFeuilles}) {$6$};
35 \node[feuillescore] (Rafa) at ({\NiveauD},{(5)*\InterFeuilles}) {$12$};
36 \node[noeud] (Rb) at ({\NiveauB},{(8.5)*\InterFeuilles}) {Pile};
37 \node[noeud] (Rba) at ({\NiveauC},{(6)*\InterFeuilles}) {$1$};
38 \node[feuillescore] (Rbaa) at ({\NiveauD},{(6)*\InterFeuilles}) {$5$};
39 \node[noeud] (Rbb) at ({\NiveauC},{(7)*\InterFeuilles}) {$2$};
40 \node[feuillescore] (Rbba) at ({\NiveauD},{(7)*\InterFeuilles}) {$6$};
41 \node[noeud] (Rbc) at ({\NiveauC},{(8)*\InterFeuilles}) {$3$};
42 \node[feuillescore] (Rbca) at ({\NiveauD},{(8)*\InterFeuilles}) {$7$};
43 \node[noeud] (Rbd) at ({\NiveauC},{(9)*\InterFeuilles}) {$4$};
44 \node[feuillescore] (Rbda) at ({\NiveauD},{(9)*\InterFeuilles}) {$8$};
45 \node[noeud] (Rbe) at ({\NiveauC},{(10)*\InterFeuilles}) {$5$};
46 \node[feuillescore] (Rbea) at ({\NiveauD},{(10)*\InterFeuilles}) {$9$};
47 \node[noeud] (Rbf) at ({\NiveauC},{(11)*\InterFeuilles}) {$6$};
48 \node[feuillescore] (Rbfa) at ({\NiveauD},{(11)*\InterFeuilles}) {$10$};
49 % Arcs (MODIFIABLES : Styles)
50 \draw[fleche] (R)--(Ra) node[etiquette] {$\dfrac{1}{2}$};
51 \draw[fleche] (Ra)--(Raa) node[etiquette] {$\frac{1}{6}$};
52 \draw[score] (Raa)--(Raaa);
53 \draw[fleche] (Ra)--(Rab) node[etiquette] {$\frac{1}{6}$};
54 \draw[score] (Rab)--(Raba);
55 \draw[fleche] (Ra)--(Rac) node[etiquette] {$\frac{1}{6}$};
56 \draw[score] (Rac)--(Raca);
57 \draw[fleche] (Ra)--(Rad) node[etiquette] {$\frac{1}{6}$};
58 \draw[score] (Rad)--(Rada);
59 \draw[fleche] (Ra)--(Rae) node[etiquette] {$\frac{1}{6}$};
60 \draw[score] (Rae)--(Raea);
61 \draw[fleche] (Ra)--(Raf) node[etiquette] {$\frac{1}{6}$};
62 \draw[score] (Raf)--(Rafa);
63 \draw[fleche] (R)--(Rb) node[etiquette] {$\dfrac{1}{2}$};
64 \draw[fleche] (Rb)--(Rba) node[etiquette] {$\frac{1}{6}$};
65 \draw[score] (Rba)--(Rbaa);
66 \draw[fleche] (Rb)--(Rbb) node[etiquette] {$\frac{1}{6}$};
67 \draw[score] (Rbb)--(Rbba);

```

```

68 \draw[fleche] (Rb)--(Rbc) node[etiquette] {$\frac{1}{6}$};
69 \draw[score] (Rbc)--(Rbca);
70 \draw[fleche] (Rb)--(Rbd) node[etiquette] {$\frac{1}{6}$};
71 \draw[score] (Rbd)--(Rbda);
72 \draw[fleche] (Rb)--(Rbe) node[etiquette] {$\frac{1}{6}$};
73 \draw[score] (Rbe)--(Rbea);
74 \draw[fleche] (Rb)--(Rbf) node[etiquette] {$\frac{1}{6}$};
75 \draw[score] (Rbf)--(Rbfa);
76 \end{tikzpicture}
77 \end{center}
78 %:-+--+--+ Fin

```



Graphes probabilistes

```

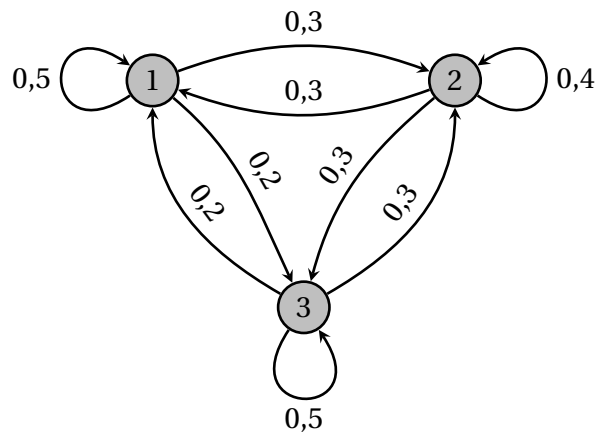
1 \begin{center}
2 \begin{tikzpicture}[scale=1]
3 \tikzstyle{every path}=[>=stealth,->,line width=1pt];

```

```

4 \node [draw,circle,fill=gray!50] (A) at (0,0) {1};
5 \node [draw,circle,fill=gray!50] (B) at (4,0) {2};
6 \node [draw,circle,fill=gray!50] (C) at (2,-3) {3};
7 \draw[->] (A) to [out=20,in=160] node[midway,sloped,above] {0,3}(B);
8 \draw[->] (A) to [out=-40,in=115] node[midway,sloped,above] {0,2}(C);
9 \draw[->] (B) to [out=-160,in=-20] node[midway,sloped,above] {0,3}(A);
10 \draw[->] (B) to [out=-140,in=75] node[midway,sloped,above] {0,3}(C);
11 \draw[->] (C) to [out=150,in=-90] node[midway,sloped,above] {0,2}(A);
12 \draw[->] (C) to [out=30,in=-90] node[midway,sloped,above] {0,3}(B);
13 \draw[->] (B) .. controls +(1.5,-1) and +(1.5,1) .. node[midway,right]{$0,4$} (B) ;
14 \draw[->] (A) .. controls +(-1.5,-1) and +(-1.5,1) .. node[midway,left]{$0,5$} (A) ;
15 \draw[->] (C) .. controls +(-1,-1.5) and +(1,-1.5) .. node[midway,below]{$0,5$} (C) ;
16 \end{tikzpicture}
17 \end{center}

```



IV Exemples en ligne



On retrouve énormément de figures créées avec TikZ sur le site texample.net.

V Exemples de figures utilisant TkZ



Les *packages* TikZ et TkZ sont compatibles et leurs commandes peuvent donc être utilisées pour une même figure.

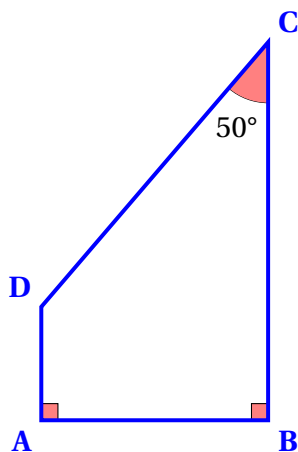
Voici une figure du plan toute simple avec des angles marqués.

Avec des angles marqués

```

1 \begin{center}
2 \begin{tikzpicture}[scale=1]
3 \tkzDefPoint(0,0){A}\tkzLabelPoints[font=\boldmath,below left,color=blue](A)
4 \tkzDefPoint(3,0){B}\tkzLabelPoints[font=\boldmath,below right,color=blue](B)
5 \tkzDefPoint(3,5){C}\tkzLabelPoints[font=\boldmath,above right,color=blue](C)
6 \tkzDefPoint(0,1.5){D}\tkzLabelPoints[font=\boldmath,above left,color=blue](D)
7 \tkzMarkRightAngle[fill=red!50](B,A,D)
8 \tkzMarkRightAngle[fill=red!50](C,B,A)
9 \tkzMarkAngle[fill=red!50,size=0.8cm](D,C,B)
10 \tkzLabelAngle[pos=1.2](D,C,B){50\degres}
11 \tkzDrawPolygon[line width=1.5pt,color=blue](A,B,C,D)
12 \end{tikzpicture}
13 \end{center}

```



Voici une figure correspond au patron d'un prisme droit et sur laquelle on a marqué des angles droits, des segments de même longueur, des arcs de cercle.

Avec des arcs de cercle

```

1 \begin{center}
2 \begin{tikzpicture}[scale=1]
3
4 \tkzDefPoint(0,0){A}\tkzLabelPoints[font=\boldmath,below left,color=blue](A)
5 \tkzDefPoint(2,0){C}\tkzLabelPoints[font=\boldmath,below right,color=blue](C)
6 \tkzDefPoint(5,0){B}\tkzLabelPoints[font=\boldmath,below right,color=blue](B)
7 \tkzDefPoint(0,4){D}\tkzLabelPoints[font=\boldmath,above left,color=blue](D)
8 \tkzDefPoint(2,4){F}\tkzLabelPoints[font=\boldmath,above right,color=blue](F)
9 \tkzDefPoint(5,4){E}\tkzLabelPoints[font=\boldmath,above right,color=blue](E)
10 \tkzDefPoint(2,7){E_1}\tkzLabelPoints[font=\boldmath,above right,color=blue](E_1)
11 \tkzDefPoint(2,-3){B_1}\tkzLabelPoints[font=\boldmath,below right,color=blue](B_1)
12
13 \tkzCalcLength[cm](A,B_1)\tkzGetLength{dAB}
14
15 \tkzDefPoint(-\dAB,0){B_2}\tkzLabelPoints[font=\boldmath,below left,color=blue](B_2)

```

```

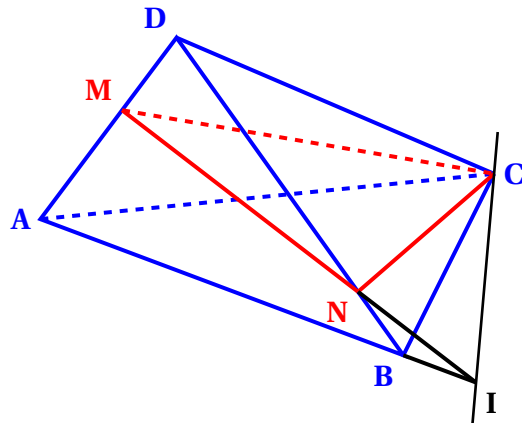
16 \tkzDefPoint(-\dAB,4){E_2}\tkzLabelPoints[font=\boldmath,above left,color=blue](E_2)
17
18 \tkzDrawArc[delta=10,line width=1.5pt,color=red,style=dashed](A,B_2)(B_1)
19 \tkzDrawArc[delta=10,line width=1.5pt,color=red,style=dashed](D,E_1)(E_2)
20
21 \tkzMarkRightAngle[fill=red!50](E_1,F,D)
22 \tkzMarkRightAngle[fill=red!50](F,C,A)
23 \tkzMarkRightAngle[fill=red!50](A,D,F)
24 \tkzMarkRightAngle[fill=red!50](D,A,B_2)
25 \tkzMarkRightAngle[fill=red!50](B_2,E_2,D)
26 \tkzMarkRightAngle[fill=red!50](C,F,E)
27 \tkzMarkRightAngle[fill=red!50](E,B,C)
28
29 \tkzMarkRightAngle[fill=blue!50](D,F,C)
30 \tkzMarkRightAngle[fill=blue!50](A,C,B_1)
31 \tkzMarkRightAngle[fill=blue!50](C,A,D)
32 \tkzMarkRightAngle[fill=blue!50](E_2,D,A)
33 \tkzMarkRightAngle[fill=blue!50](A,B_2,E_2)
34 \tkzMarkRightAngle[fill=blue!50](B,C,F)
35 \tkzMarkRightAngle[fill=blue!50](F,E,B)
36
37 \tkzDrawPolygon[line width=1.5pt,color=blue](B_2,A,D,E_2)
38 \tkzDrawPolygon[line width=1.5pt,color=blue](A,C,F,D)
39 \tkzDrawPolygon[line width=1.5pt,color=blue](C,B,E,F)
40 \tkzDrawPolygon[line width=1.5pt,color=blue](A,C,B_1)
41 \tkzDrawPolygon[line width=1.5pt,color=blue](D,F,E_1)
42
43 \tkzMarkSegments[mark=||,size=4pt](B_1,C C,B F,E F,E_1)
44 \tkzMarkSegments[mark=o,size=4pt](E_2,B_2 A,D C,F B,E)
45 \tkzMarkSegments[mark=|||,size=4pt](A,B_1 A,B_2 D,E_1 D,E_2)
46 \tkzMarkSegments[mark=oo,size=8pt](A,C D,F)
47
48
49 \end{tikzpicture}
50 \end{center}

```

```

22
23 \tkzDrawSegments[line width=1.5pt,color=red](M,N N,C)
24 \tkzDrawSegments[line width=1.5pt,color=red,style=dashed](M,C)
25
26 \tkzDrawSegments[line width=1.5pt](N,I B,I)
27 \tkzDrawLine[line width=1pt](I,C)
28
29 \end{tikzpicture}
30 \end{center}

```



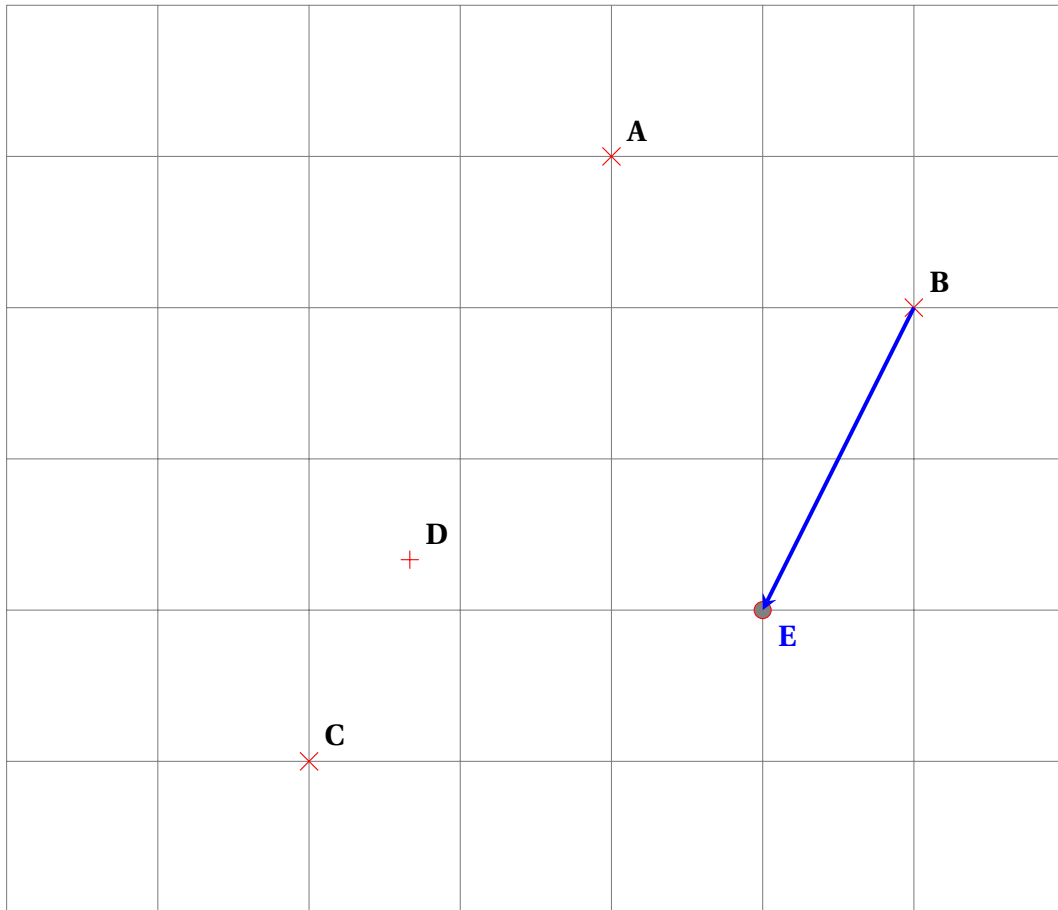
Voici une figure avec des vecteurs.

Avec des vecteurs

```

1 \begin{center}
2 \begin{tikzpicture}[scale=2,>stealth]
3 \tkzInit[xmin=-2,xmax=5,ymin=-2,ymax=4]
4 \tkzGrid
5 \tkzDefPoint(2,3){A}\tkzDefPoint(4,2){B}
6 \tkzDefPointWith[orthogonal,K=-2](A,B)\tkzGetPoint{C}%AC=2*AB et (AB,AC)=-pi/2
7 \tkzDefPointWith[linear,K=2/3](A,C)\tkzGetPoint{D}%\vect{AD}=2/3*\vect{AC}
8 \tkzDefPointWith[colinear=at B,K=1/2](A,C)\tkzGetPoint{E}%\vect{BE}=1/2*\vect{AC}
9 \tkzDrawPoints[shape=cross out,color=red,size=16pt](A,B,C)
10 \tkzDrawPoints[shape=cross,color=red,size=16pt](D)
11 \tkzDrawPoints[color=red,size=16pt](E)
12 \tkzLabelPoints[above right=3pt,font=\boldmath](A,B,C,D)
13 \tkzLabelPoints[below right=3pt,font=\boldmath,color=blue](E)
14 \tkzDrawSegment[->,color=blue,line width=1.5pt](B,E)
15 \end{tikzpicture}
16 \end{center}

```



Voici un polygone d'effectifs cumulés croissants.

Polygone d'E.C.C.

```

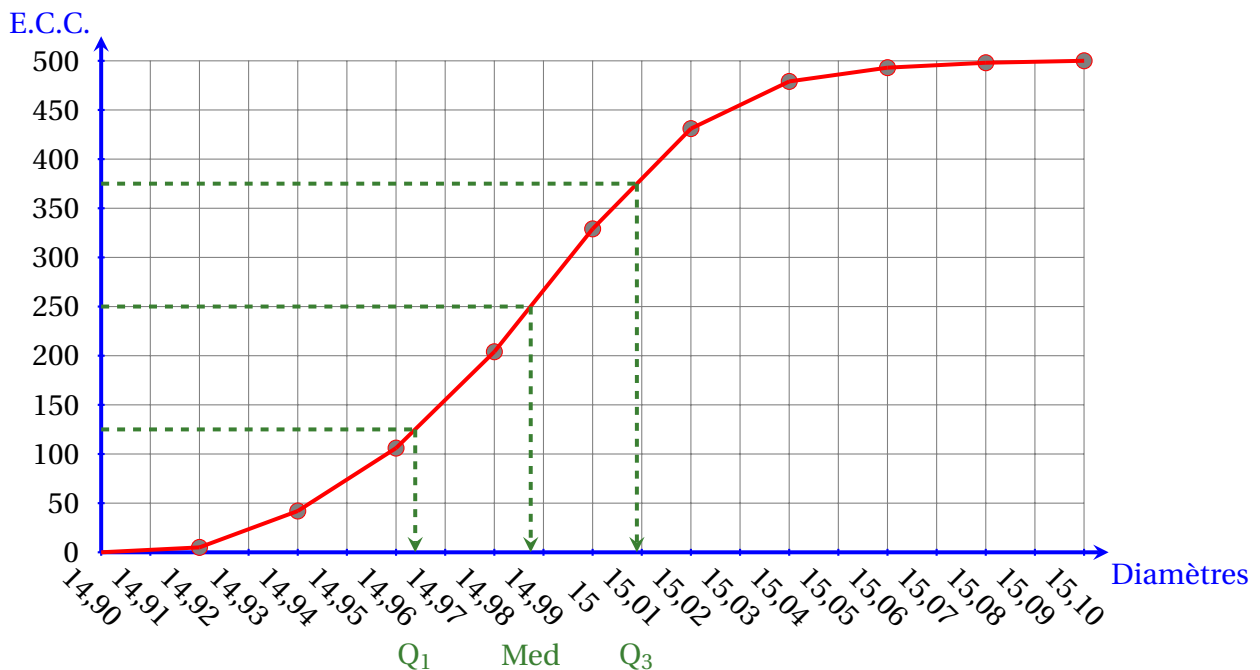
1 \begin{center}
2 \begin{tikzpicture}[scale=0.65,>=stealth]
3 \tkzInit[xmin=14.90,xmax=15.10,xstep=0.01,ymin=0,ymax=500,ystep=50]
4 \tkzGrid
5 \tkzDrawX[line width=1.5pt,color=blue,below right,label=Diamètres,>=stealth]
6 \tkzLabelX[label options={below=12pt,rotate=-45}]
7 \tkzDrawY[line width=1.5pt,color=blue,above left,label=E.C.C.,>=stealth]
8 \tkzLabelY[label options={left=6pt}]
9 \tkzDefPoint(14.9,0){A}
10 \tkzDefPoint(14.92,5){B}
11 \tkzDefPoint(14.94,42){C}
12 \tkzDefPoint(14.96,106){D}
13 \tkzDefPoint(14.98,204){E}
14 \tkzDefPoint(15,329){F}
15 \tkzDefPoint(15.02,431){G}
16 \tkzDefPoint(15.04,479){H}
17 \tkzDefPoint(15.06,493){I}
18 \tkzDefPoint(15.08,498){J}
19 \tkzDefPoint(15.1,500){K}
20 \tkzDrawPoints[color=red,size=4pt](B,C,D,E,F,G,H,I,J,K)
21 \tkzDrawSegments[color=red,line width=1.5pt](A,B B,C C,D D,E E,F F,G G,H H,I I,J J,K)
22 %Médiane

```

```

23 \tkzDefPoint(14.9,250){L}
24 \tkzDefPoint(14.9874,250){M}
25 \tkzDefPoint(14.9874,0){N}
26 \tkzDrawSegment[color=OliveGreen,line width=1.5pt,style=dashed](L,M)
27 \tkzDrawSegment[->,>=stealth,color=OliveGreen,line width=1.5pt,style=dashed](M,N)
28 \tkzText[below=30pt,color=OliveGreen](N){Med}
29 %1er quartile
30 \tkzDefPoint(14.9,125){R}
31 \tkzDefPoint(14.9639,125){S}
32 \tkzDefPoint(14.9639,0){T}
33 \tkzDrawSegment[color=OliveGreen,line width=1.5pt,style=dashed](R,S)
34 \tkzDrawSegment[->,>=stealth,color=OliveGreen,line width=1.5pt,style=dashed](S,T)
35 \tkzText[below=30pt,color=OliveGreen](T){$Q_1$}
36 %3eme quartile
37 \tkzDefPoint(14.9,375){U}
38 \tkzDefPoint(15.009,375){V}
39 \tkzDefPoint(15.009,0){W}
40 \tkzDrawSegment[color=OliveGreen,line width=1.5pt,style=dashed](U,V)
41 \tkzDrawSegment[->,>=stealth,color=OliveGreen,line width=1.5pt,style=dashed](V,W)
42 \tkzText[below=30pt,color=OliveGreen](W){$Q_3$}
43 \end{tikzpicture}
44 \end{center}

```



Voici une figure construite sur un quadrillage Seyes avec des vecteurs.

Cette figure utilise une commande qui a été créée dans le preambule et qui est la suivante :

```

\newcommand{\quadrillageSeyes}[2]
{\draw[line width=0.3mm, color=blue!10, ystep=0.2, xstep=0.8] #1 grid #2;
\draw[line width=0.3mm, color=blue!30, xstep=0.8, ystep=0.8] #1 grid #2; }

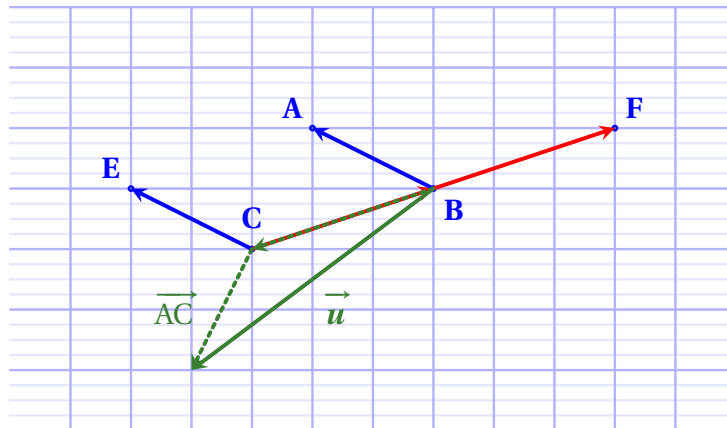
```

Quadrillage Seyes et vecteurs

```

1 \begin{center}
2 \begin{tikzpicture}
3 [line width=0.3mm, >=stealth, x=1cm, y=1cm, line cap=round, line join=round, scale=1]
4 \quadrillageSeyes{(-3.2,-2.4)}{(6.4,3.2)}
5 \tkzDefPoint(0.8,1.6){A}\tkzLabelPoints[font=\boldmath,above left,color=blue](A)
6 \tkzDefPoint(2.4,0.8){B}\tkzLabelPoints[font=\boldmath,below right,color=blue](B)
7 \tkzDefPoint(0,0){C}\tkzLabelPoints[font=\boldmath,above=4pt,color=blue](C)
8 \tkzDefPointWith[colinear=at C,K=1](B,A)\tkzGetPoint{E}%\vect{CE}=\vect{BA}
9 \tkzLabelPoints[font=\boldmath,above left,color=blue](E)
10 \tkzDefPointWith[colinear=at B,K=1](C,B)\tkzGetPoint{F}%\vect{BF}=\vect{CB}
11 \tkzLabelPoints[font=\boldmath,above right,color=blue](F)
12 \tkzDrawPoints[size=2,color=blue](A,B,C,E,F)
13 \tkzDrawVectors[color=blue,line width=1.5pt,>=stealth](C,E B,A)
14 \tkzDrawVectors[color=red,line width=1.5pt,>=stealth](B,F C,B)
15
16 \tkzDefPointWith[colinear=at C,K=1](A,C)\tkzGetPoint{G}%\vect{CG}=\vect{AC}
17 \tkzDrawVectors[color=OliveGreen,line width=1.5pt,style=dotted,>=stealth](B,C C,G)
18 \tkzDrawVector[color=OliveGreen,line width=1.5pt,>=stealth](B,G)
19 \tkzLabelSegment[below right,color=OliveGreen,font=\boldmath](B,G){$\vect{u}$}
20 \tkzLabelSegment[left=4pt,color=OliveGreen](C,G){$\vect{AC}$}
21
22 \end{tikzpicture}
23 \end{center}

```



Voici un graphique comportant des tangentes et un effet loupe.
 Pour l'effet loupe, la librairie *spy* de TikZ a besoin d'être chargée.

Tangentes et effet loupe

```

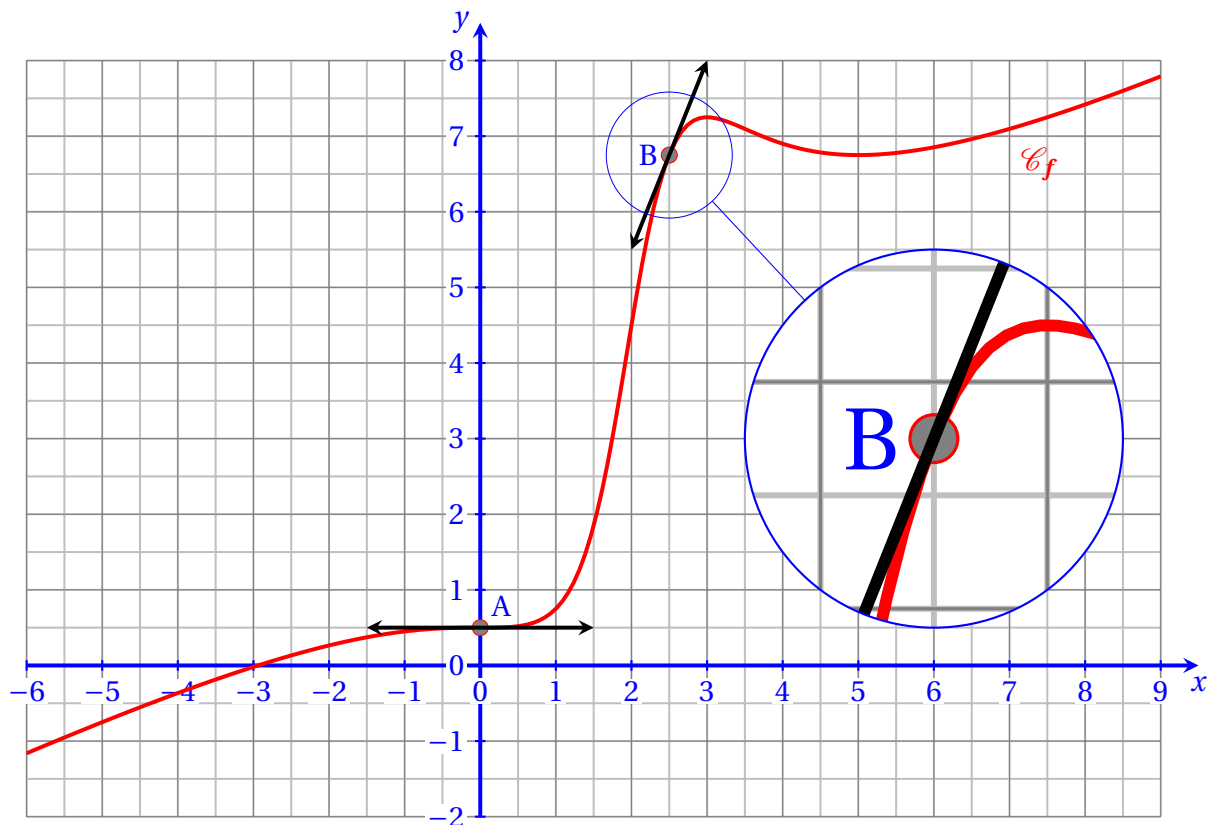
1 % requires \usetikzlibrary{spy}
2
3 \begin{center}
4 \begin{tikzpicture}[scale=1,spy using outlines={circle,magnification=3,connect spies}]
5 \tkzInit[xmin=-6,xmax=9,ymin=-2,ymax=8]
6 \tkzGrid[sub,subxstep=0.5,subystep=0.5]

```

```

7 \tkzAxeXY[>=stealth,color=blue,line width=1.5pt]
8 \tkzClip
9 %Courbe de la fonction
10 \tkzFct[color=red,samples=200,line width=1.5pt]
11 {(\x**3+\x**2-4*\x+5)/(2*\x**2-8*\x+10)}
12 \tkzDefPoint(7,7){F}
13 \tkzText[font=\boldmath,color=red,below right](F){$\cal C_f$}
14
15 %Points du graphique
16 \tkzDefPoint(0,0.5){A}\tkzLabelPoints[font=\bfseries,above right,color=blue](A)
17 \tkzDefPoint(2.5,6.75){B}\tkzLabelPoints[font=\bfseries,left,color=blue](B)
18 \tkzDrawPoints[size=4,color=red](A,B)
19
20 %Tangentes tracées
21 \tkzDefPoint(3,8){C}
22 \tkzDefPointWith[linear,K=2](C,B)\tkzGetPoint{G}%\vect{CG}=2*\vect{CB}
23 \tkzDrawSegment[<->,>=stealth,line width=1.5pt](C,G)
24 \tkzDefPoint(-1.5,0.5){D}
25 \tkzDefPoint(1.5,0.5){E}
26 \tkzDrawSegment[<->,>=stealth,line width=1.5pt](D,E)
27
28 %effet loupe
29 \tkzDefPoint(6,3){magnifyglass}
30 \spy [blue, size=5cm] on (B) in node[fill=white] at (magnifyglass);
31 \end{tikzpicture}
32 \end{center}

```





Avec la commande `\tkzfct`, utilisée dans l'exemple précédent, on peut tracer des courbes de fonctions avec une syntaxe relativement simplifiée.

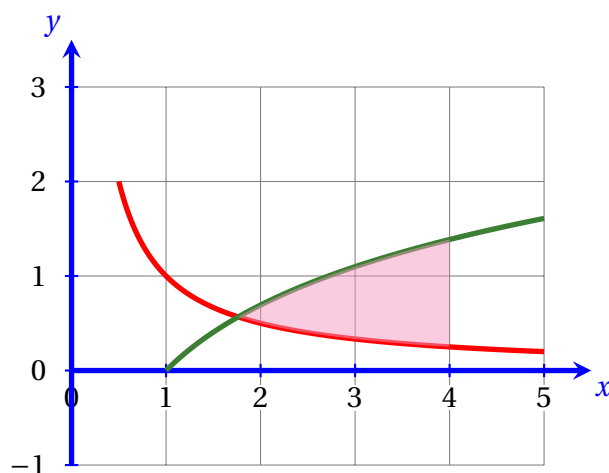
Voici d'autres exemples :

Aire entre deux courbes

```

1 \begin{center}
2 \begin{tikzpicture}[scale=1.25]
3 \tkzInit[ymin=-1,xmax=5,ymax=3]
4 \tkzGrid
5 %\tkzAxeXY
6 \tkzDrawX[line width=2pt,color=blue,below right,>=stealth]%,label=Rang année]
7 \tkzLabelX%[label options={below=12pt,rotate=-45}]
8 \tkzDrawY[line width=2pt,color=blue,above left,>=stealth]%,label=Montant (en \euro)]
9 \tkzLabelY[label options={left=6pt}]
10 \tkzFct[domain = 0.5:5,line width=2pt,color=red]{1/x}% courbe a
11 \tkzFct[domain = 1:5,line width=2pt,color=OliveGreen]{log(x)}% courbe b
12 \tkzDrawAreafg[between=b and a,
13 color=magenta!50,
14 domain = 1:4]
15 \end{tikzpicture}
16 \end{center}

```



Visualisation d'une somme de Riemann.

Une somme de Riemann

```

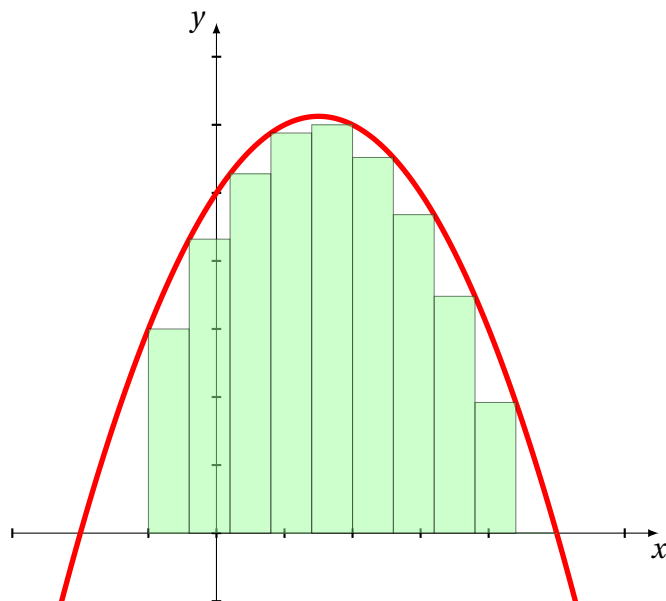
1 \begin{center}
2 \begin{tikzpicture}[scale=0.9]
3 \tkzInit[xmin=-3,xmax=6,ymin=-2,ymax=14,ystep=2]
4 \tkzDrawX \tkzDrawY
5 \tkzFct[line width=2pt,color = red, domain = -3:6]{(-\x-2)*(\x-5)}

```

```

6 \tkzDrawRiemannSumInf[fill=green!40,opacity=.5,interval=-1:5,number=10]
7 \end{tikzpicture}
8 \end{center}

```



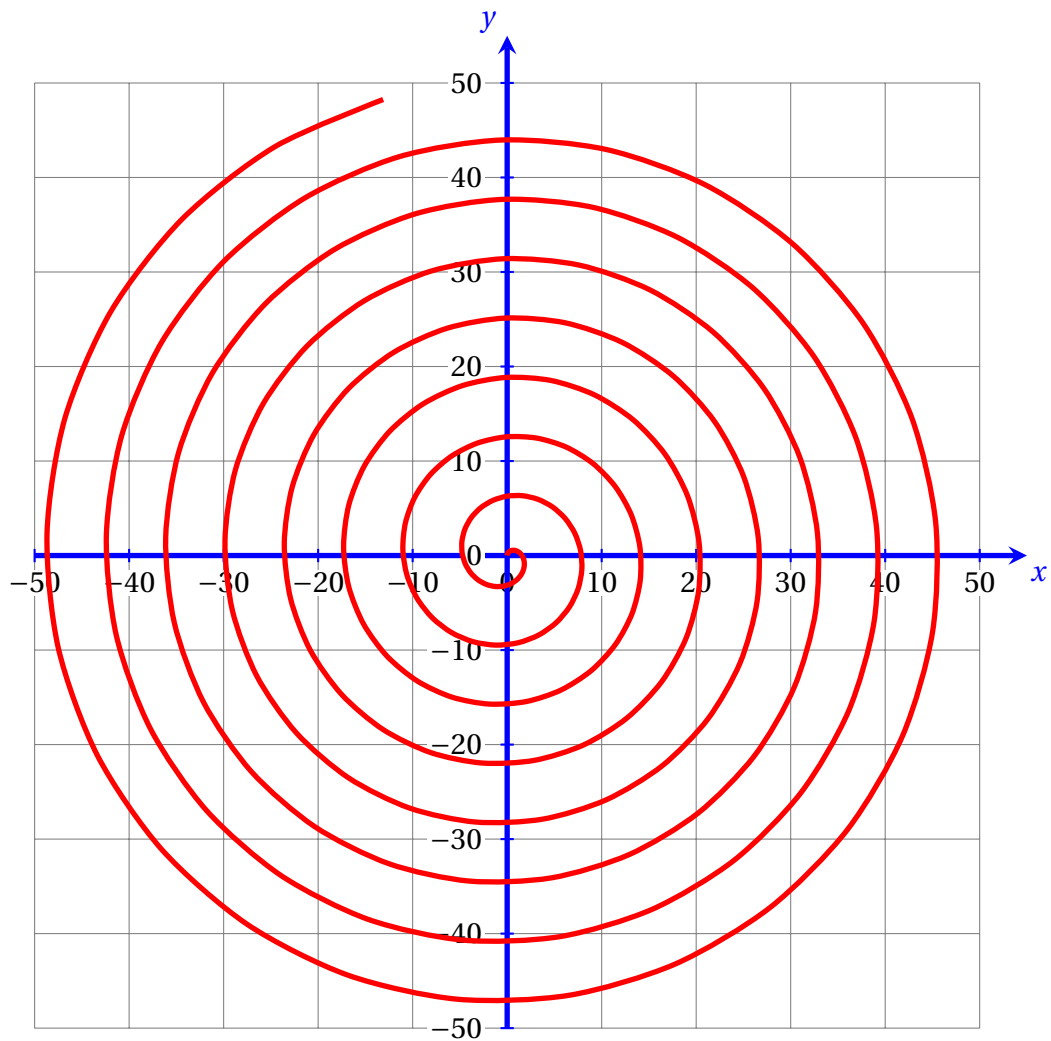
Une courbe paramétrée.

Une courbe paramétrée

```

1 \begin{center}
2 \begin{tikzpicture}[scale=1.25]
3 \tkzInit[xmin=-50,xmax=50,xstep=10,
4 ymin=-50,ymax=50,ystep=10]
5 \tkzGrid
6 \tkzDrawX[line width=2pt,color=blue,below right,>=stealth]
7 \tkzLabelX%[label options={below=12pt,rotate=-45}]
8 \tkzDrawY[line width=2pt,color=blue,above left,>=stealth]
9 \tkzLabelY[label options={left=6pt}]
10 \tkzFctPar[color=red,line width=2pt,smooth,samples=200,domain=0:50]{t*sin(t)}{t*cos(t)}
11 \end{tikzpicture}
12 \end{center}

```



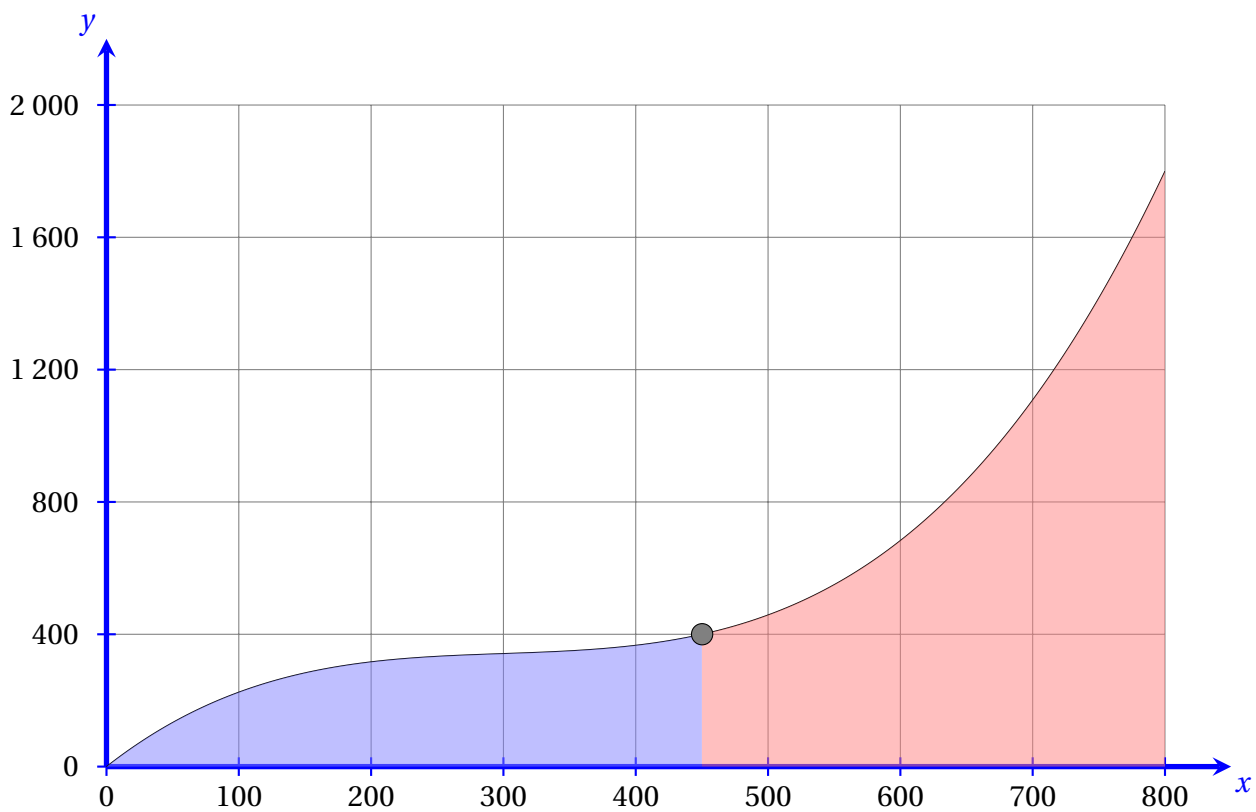
Des aires sous une courbe.

Des aires sous une courbe

```

1 \begin{tikzpicture}[scale=1.75]
2 \tkzInit[xmin=0,xmax=800,xstep=100,
3 ymin=0,ymax=2000,ystep=400]
4 \tkzGrid
5 \tkzDrawX[line width=2pt,color=blue,below right,>=stealth]
6 \tkzLabelX%[label options={below=12pt,rotate=-45}]
7 \tkzDrawY[line width=2pt,color=blue,above left,>=stealth]
8 \tkzLabelY[label options={left=6pt}]
9 \tkzFct[domain = 0:800]{(1./90000)*\x*\x*\x-(1./100)*\x*\x+(113./36)*\x}
10 \tkzDefPoint(450,400){a}
11 \tkzDrawArea[color=blue!50, domain =0:450]
12 \tkzDrawArea[color=red!50, domain =450:800]
13 \tkzDrawPoint[size=4pt](a)
14 \end{tikzpicture}

```



Arbres pondérés

On a vu dans le chapitre 7 un exemple pour obtenir des arbres pondérés.

En utilisant Pstricks, on obtient une syntaxe souvent plus simple pour obtenir de tels arbres.

En voici quelques exemples :

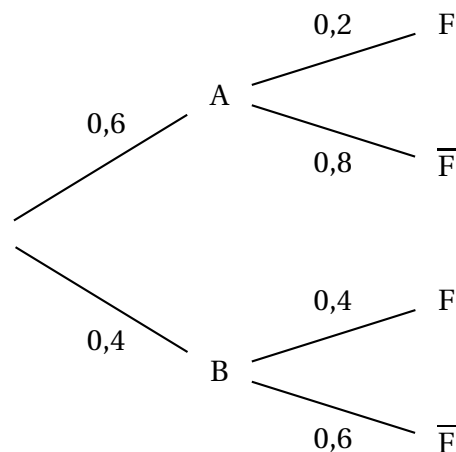
Arbre 2x2

```

1 \psset{nodesep=3mm,levelsep=30mm,treesep=15mm}
2 \begin{center}
3 \pstree[treemode=R]{\TR{}}
4 {
5   \pstree{\TR{$A$}}\taput{$0,6$}
6   {
7     \TR{$F$} \taput{$0,2$}
8     \TR{$\overline{F}$} \tbput{$0,8$}
9   }
10
11  \pstree{\TR{$B$}} \tbput{$0,4$}
12  {
13    \TR{$F$} \taput{$0,4$}
14    \TR{$\overline{F}$} \tbput{$0,6$}
15  }
16 }
17 \end{center}

```

Ce code source donne :



Arbre 2x3

```

1 \psset{nodesep=3mm,levelsep=30mm,treesep=15mm}
2 \begin{center}
3 \pstree[treemode=R]{\TR{}}
4 {
5   \pstree{\TR{$G$}}\taput{$0,3$}
6   {

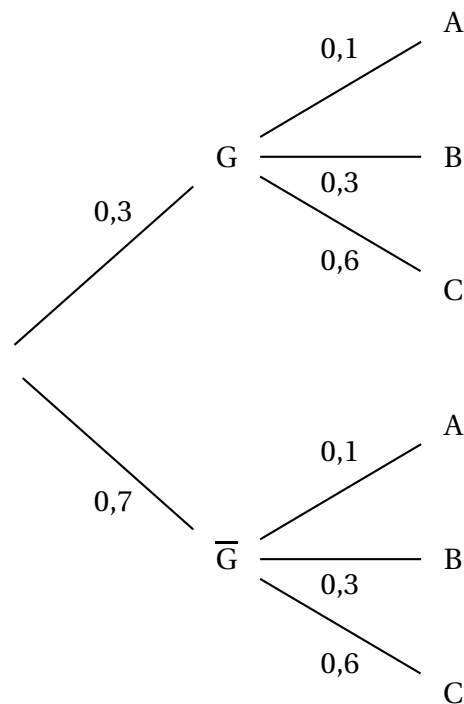
```

```

7      \TR{$A$} \taput{$0,1$}
8      \TR{$B$} \tbput{$0,3$}
9      \TR{$C$} \tbput{$0,6$}
10     }
11     \pstree{\TR{$\overline{G}$}\tbput{$0,7$}}
12     {
13       \TR{$A$} \taput{$0,1$}
14       \TR{$B$} \tbput{$0,3$}
15       \TR{$C$} \tbput{$0,6$}
16     }
17   }
18 \end{center}

```

Ce code source donne :



Arbre 3x2

```

1 \psset{nodesep=3mm,levelsep=30mm,treesep=15mm}
2 \begin{center}
3 \pstree[treemode=R]{\TR{}}
4 {
5   \pstree{\TR{$A$}\taput{$0,6$}}
6   {
7     \TR{$F$} \taput{$0,2$}
8     \TR{$\overline{F}$} \tbput{$0,8$}
9   }
10  \pstree{\TR{$B$}\taput{$0,3$}}
11  {
12    \TR{$F$} \taput{$0,3$}
13    \TR{$\overline{F}$} \tbput{$0,7$}
14  }
15  \pstree{\TR{$C$} \tbput{$0,1$}}

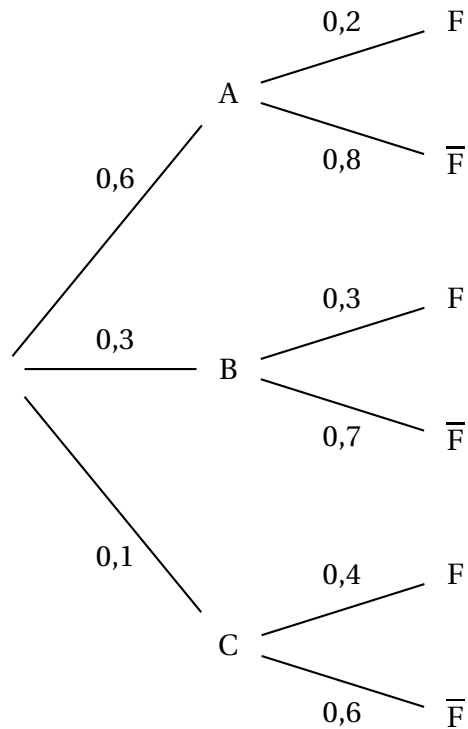
```

```

16      {
17      \TR{$F$} \taput{$0,4$}
18      \TR{$\overline{F}$} \tbput{$0,6$}
19      }
20 }
21 \end{center}

```

Ce code source donne :



Arbres non symétriques

```

1 \begin{center}
2 \pstree[treemode=R]{\TR{}}
3 {
4   \pstree{\TR{$C$}\taput{$0,3$}}
5   {
6     \TR{$B$} \taput{$0,2$}
7     \TR{$\overline{B}$} \tbput{$0,8$}
8   }
9   \TR{$\overline{C}$} \tbput{$0,7$}
10 }
11 \end{center}
12
13
14 \begin{center}
15 \pstree[treemode=R]{\TR{}}
16 {
17   \TR{$\overline{A}$} \taput{$0,7$}
18   \pstree{\TR{$B$}\tbput{$0,3$}}
19   {
20     \TR{$A$} \taput{$0,7$}

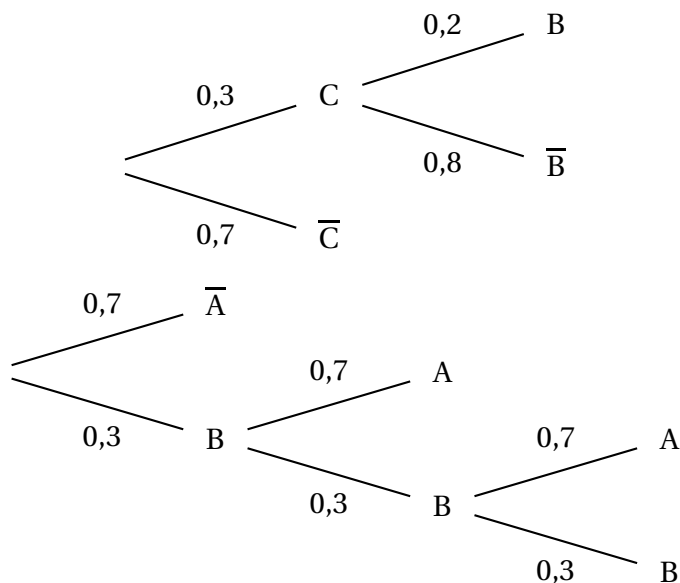
```

```

21 \pstree{\TR{$B$}\tbput{$0,3$}}
22 {
23 \TR{$A$} \taput{$0,7$}
24 \TR{$B$} \tbput{$0,3$}
25 }
26 }
27 }
28 \end{center}

```

Ce code source donne :



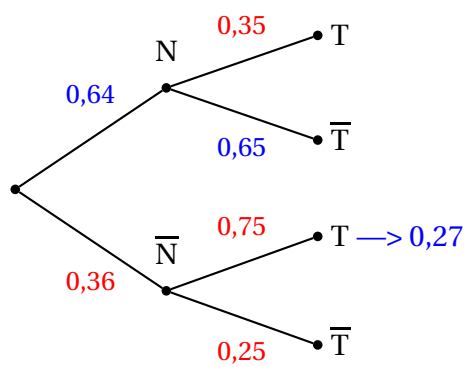
Arbre avec des points

```

1 \psset{nodesep=0mm,levelsep=20mm,treesep=10mm}
2 \pstree[treemode=R]{\Tdot}
3 {
4 \pstree
5 {\Tdot~[tnpos=a]{$N$}\taput{\small $\color{blue} 0,64$}}
6 {
7 \Tdot~[tnpos=r]{$T$}\taput{\small $\color{red} 0,35$}}
8 \Tdot~[tnpos=r]{$\overline{T}$}\tbput{\small $\color{blue} 0,65$}}
9 }
10 \pstree
11 {\Tdot~[tnpos=a]{$\overline{N}$}\tbput{\small $\color{red} 0,36$}}
12 {
13 \Tdot~[tnpos=r]{$T$ \color{blue} ---> 0,27} }\taput{\small $\color{red} 0,75$}}
14 \Tdot~[tnpos=r]{$\overline{T}$}\tbput{\small $\color{red} 0,25$}}
15 }
16 }

```

Ce code source donne :



QR codes

Il existe plusieurs *packages* permettant de générer des QR-codes.
Voici quelques exemples :

QR codes

```
1 \begin{center}
2 %Exemple nécessitant le package qrcode
3 \qrcode[hyperlink,height=2cm]{http://www.ac-orleans-tours.fr/}
4 \hfill
5 %Exemple nécessitant PsTricks
6 \begin{pspicture}(2,2)
7 \psbarcode{http://www.ac-orleans-tours.fr/}{eclevel=L}{qrcode}
8 \end{pspicture}
9 \end{center}
```

Le *code source* précédent donne :

