



1. Pour introduire le problème
2. Les solutions précédentes
3. Ma solution
4. Quelques remarques pour conclure



1. Introduction
2. Etat de l'art
3. Résultats
4. Conclusion

Pour structurer une (sous)-sous-section avec des titres, on peut utiliser des titres non-numérotés qui n'apparaissent pas dans la table des matières. C'est la méthode qui a été utilisée dans ce document (*e.g.*, la présente section).



2. Algorithme de Tarjan
  - 2.1. Idée générale
    - 2.1.1. Utilisation du DFS
      - 2.1.1.1. Rappels
 

Texte.
      - 2.1.1.2. Pseudo-code
 

Texte.



2. Algorithme de Tarjan
  - 2.1. Idée générale
    - 2.1.1. Utilisation du DFS
 

**Rappels**

Texte.

**Pseudo-code**

Texte.

## Structure interne

Les sections identifiées dans le plan de rédaction servent à donner une structure *globale* au travail écrit. Cependant, il faut également structurer *localement* le texte à l'intérieur de chaque section.

Les unités de présentation qui servent à structurer le texte dans une section sont les *paragraphes*. Ceux-ci doivent se suivre de façon logique. Un paragraphe doit également être organisé *lui-même* de façon logique. Il est constitué de quelques phrases. Ces phrases sont liées et communes à un point précis du sujet qui est l'objet de la section qui le contient.



Comme pour la structure générale du travail, n'organisez pas vos paragraphes de manière *chronologique*. Structurer la présentation des résultats et éviter de donner la liste chronologique des essais infructueux et des erreurs commises. Le premier paragraphe d'une section devrait introduire le sujet de celle-ci.



Essayez d'équilibrer les paragraphes pour qu'ils ne soient ni trop courts (une ou deux lignes), ni trop longs (une demi-page). Un paragraphe très court peut servir à isoler une idée importante, mais cela doit rester une exception.

## Introduction

L'introduction est en général organisée comme suit [3, 15] : contexte, définition du problème, présentation et limitations des solutions existantes (s'il y en a), objectifs du travail et idées principales. Elle se termine par une brève description du contenu, chapitre par chapitre (voir par exemple les deux derniers paragraphes de la Section 1).



L'introduction est une partie importante du texte. Elle doit convaincre le lecteur que le travail vaut la peine d'être lu. Il faut motiver ce lecteur, qui n'est peut-être pas *a priori* intéressé par votre travail. Expliquez pourquoi le problème étudié est important, quelle sera votre contribution et pourquoi les solutions apportées sont appropriées. Gardez à l'esprit que le lecteur n'a pas encore lu le travail, qu'il ne connaît pas le sujet et qu'il n'est pas un expert du domaine.



Dans votre introduction, parlez du *sujet* de votre travail, et pas du fait que vous réalisez un travail en informatique (les lecteurs sont au courant).

Valduriez [15] propose de *commencer* la rédaction d'un article scientifique par l'introduction. Cela permet de fixer les motivations et la structure. Par contre, Bruyère [3] propose d'écrire l'introduction de travaux d'étudiants *au terme* de la rédaction. Il est en effet plus facile de l'écrire une fois le travail réalisé et son contenu fixé. Bien que la rédaction d'un article scientifique ne soit pas identique à celle d'un mémoire, les deux approches ont leur avantage.

## Chapitres et sections

La rédaction des chapitres et des sections constitue le coeur du travail. On y présente l'objet et le développement du travail selon le plan constitué au préalable. Il faut développer les idées principales et les résultats et convaincre le lecteur de leur importance et de leur validité.

Le contenu des chapitres doit suivre une approche scientifique. Quelques conseils concernant la manière de proposer un contenu rigoureux sont donnés dans la Section 4.2.



Quand vous rédigez une section, gardez en tête que le titre de celle-ci – qui sert à structurer le texte et à introduire le sujet de la section – ne sera pas toujours lu par le lecteur ! Le premier paragraphe de la section devrait l'introduire en précisant son sujet car seul le titre ne suffit pas. Pour des sections de haut niveau (comme un chapitre), il est utile de commencer par une brève description du contenu, en présentant les sous-sections. Remarquez que cette technique a été appliquée au présent document pour les sections de plus haut niveau.

## Conclusion

La conclusion est la dernière partie du travail écrit (la bibliographie et les annexes n'étant pas considérées comme faisant partie du texte lui-même). Elle est en général organisée comme suit [3, 15] : résumé du travail et des contributions, rappel des résultats principaux, applications possibles des résultats (s'il y a lieu), limitations de la solution proposée et perspectives (pistes pour d'éventuels travaux futurs).

Le texte de la conclusion doit rester neutre mais doit mettre en avant l'apport de l'auteur par rapport au sujet.





La conclusion est une partie importante du texte. Elle résume la contribution personnelle du travail et met en avant les principaux résultats. Elle sert à remettre en place tout ce qui précède. Contrairement à l'introduction où le lecteur ne connaît encore rien du sujet, ici on suppose que le lecteur a lu l'ensemble du travail. La conclusion permet au lecteur de confirmer son opinion sur l'étendue du travail réalisé.

Il est conseillé de rédiger la conclusion en dernier lieu [3, 15].

## Autres parties du travail


Outre les parties vues ci-dessus, un travail écrit comporte également les parties suivantes (cf. Tableau 2, p. 4) :


- les *annexes* (facultatif) se placent à la fin du rapport. Elles regroupent les éléments qui ne sont pas indispensables à la compréhension du travail (la lecture des annexes est optionnelle). Celles-ci sont présentes dans le souci d’avoir un rapport complet. On y placera par exemple l’implémentation d’un algorithme qui a été présenté dans le corps du travail ou une description de la syntaxe des langages de programmation utilisés dans le texte ;
  -  Ne mettez pas l’entièreté du code source dans les annexes si celui-ci est long. Un CD-ROM annexé au document – contenant votre code, les données de tests ou les logiciels utilisés – sera plus utile et écologique.
- la *couverture* qui doit contenir : le nom de l’institut ou du département et de l’université dans lequel le travail a été réalisé ; les logos de l’université et de l’académie ; le nom de l’auteur ; le titre du travail ; la date de remise du travail (mois, année) et le nom du directeur. Pour un mémoire, il faut inclure la mention « Mémoire présenté en vue de l’obtention du grade académique de (...) » ;
  -  Toutes les informations reprises ci-dessus doivent être présentes sur la couverture. N’intitulez pas votre travail « mémoire de fin d’études » ou « Pré-rapport du projet ». Donnez un *vrai titre* à votre travail, qui décrit précisément votre sujet, comme « Arbres binaires de recherches équilibrés ». Vous pouvez cependant indiquer sur la couverture qu’il s’agit d’un pré-rapport ou autre, mais cela ne constitue pas le titre de votre travail.
- la *page de garde* (page blanche qui suit la couverture), qui est souvent suivie d’une page identique à la couverture ;
- la *table des matières*, nécessaire pour un travail volumineux, se place au début du travail (ou plus rarement à la fin) ;
- les *remerciements* (facultatif) mentionnent les personnes qui vous ont aidées (directeur, relecteurs, rapporteurs, *etc.*). Ils se placent au début (*e.g.*, mémoire) ou en fin du travail (*e.g.*, article ou rapport).


## 4.2 Contenu


Pour convaincre le lecteur, une approche scientifique doit être suivie (hypothèses, mesures, vérification, preuve, *etc.*). Toute affirmation ou donnée chiffrée doit être justifiée ou, si elle n’est pas personnelle, doit être citée en incluant une référence. Le lecteur doit pouvoir clairement faire la distinction entre ce qui est un apport personnel et ce qui ne l’est pas (*cf.* Section 3.3).


Discutez toujours les choix technologiques et les alternatives possibles. Pourquoi utiliser tel ou tel outil, langage, algorithme, formalisme ? Définissez et utilisez des critères précis et objectifs pour motiver le choix effectué (complexité en notation grand- $O$ , résultats de tests et temps CPU, caractéristiques, *etc.*).


 En évolution logicielle, seulement 30 à 40% des classes réellement modifiées sont identifiées comme susceptibles de l’être.

 Nous avons choisi d’utiliser le langage  $X$  car nous l’avons déjà utilisé dans le cadre du cours  $A$ .

 Sur base d’une étude empirique, Lindvall et Sandahl [8] observent que seulement 30 à 40% des classes réellement modifiées avaient été identifiées comme susceptibles de l’être.

 Une des caractéristiques du langage  $X$  est de pouvoir utiliser l’héritage multiple, ce qui n’est pas le cas des langages  $Y$  et  $Z$ . L’héritage multiple est important dans notre cas, car (...).

 Pour ma part, je crois que le TCL et le TK sont des langages fantastiques, permettant énormément de choses sans devoir lire dix livres et qui ne nécessitent pas dix ans de pratique.

 Entre les deux algorithmes précédemment décrits, ce fut *A* qui fut retenu, et ce, pour diverses raisons. Ce fut tout d'abord pour permettre une manipulation plus simple de la structure de donnée. Enfin, l'algorithme *A* est plus « objectif » que le *B*. Or, il me semblait nécessaire d'être objectif.




Fantastiques ? Enormément de choses ? Dix livres ? Dix ans de pratique ? Quels sont les critères objectifs en faveur de ces langages ? Soyez neutre et quantifiez avec précision.



Les complexités en temps dans le *pire des cas* des deux algorithmes *A* et *B* sont identiques ( $O(n^2)$ ). Cependant, les temps d'exécution *en moyenne* sont différents :  $\Theta(n \log n)$  pour l'algorithme *A* et  $\Theta(n^2)$  pour *B*. On peut donc espérer, sur des données aléatoires, de meilleurs temps d'exécution en choisissant l'algorithme *A*. Cette espérance théorique est confirmée par les tests présentés ci-dessous. (...)

## Tests sur ordinateurs

Dans le cas d'une expérience sur ordinateur, il faut toujours mentionner la configuration de la machine (*hardware*). Il faut également, pour *tous* les logiciels utilisés lors de l'expérimentation, indiquer d'où ils viennent (site web pour les télécharger) et quelle version a été utilisée. Ces éléments permettent au lecteur de reproduire lui-même l'expérience.

 Le temps d'exécution du programme sur les données *X* est de 12 secondes et sur les données *Y* de 23 secondes.



Le Tableau *x* reprend les temps d'exécution, exprimés en micro-secondes CPU, sur les différents ensembles de données. Ces tests ont été effectués sur une machine ayant les caractéristiques suivantes : Dell Dual Core, 2.66 GHz, 2 Gb RAM, système SuSE Linux 10.0 (kernel 2.4.2), java 1.5.0, etc. Pour calculer le temps CPU, la classe *ThreadMXBean* a été utilisée.

## Algorithmes

Pour que le lecteur puisse facilement comprendre un algorithme complexe, seul le pseudo-code ne suffit pas. Il faut utiliser une approche qui présente les grandes idées, avant de détailler les choses progressivement. Par exemple, pour un algorithme important, on peut :

1. présenter l'objectif de l'algorithme (entrées, sorties) ;
2. donner les grandes idées de son fonctionnement en français ;
3. détailler les idées importantes ;
4. donner le pseudo-code (si celui-ci est trop long, les parties importantes seulement) ;
5. appliquer l'algorithme sur un exemple ;
6. prouver son exactitude ;
7. donner et prouver sa complexité en temps et en mémoire ;
8. éventuellement, donner une implémentation en annexe.



Pour présenter les algorithmes ou les structures de données, inspirez-vous de livres de références en informatique, comme celui d'Aho et Ullman [1] ou de Cormen *et al.* [4].


### 4.3 Style


Vous n'écrivez pas un roman mais un document scientifique. Cela implique un style adapté. Le texte doit être compréhensible par un non-spécialiste du sujet<sup>7</sup>. Il faut éviter le style « prise de notes », faire des phrases et s'inspirer du style d'un livre scientifique. Un style propre à un travail scientifique devrait respecter les consignes suivantes [3, 14, 15].

1. *Précision*. Il faut définir précisément les notions (*formalisme*) la première fois qu'ils apparaissent et toujours utiliser le même terme pour y référer (*cohérence*). Quand on introduit pour la première fois un concept, on le note en italique. Il en va de même pour les notations
2. *Concision*. Allez à l'essentiel et faites des phrases courtes. Evitez d'utiliser des mots inutiles. Essayez de ne donner qu'une seule idée par phrase. Une phrase complexe peut être coupée en phrases plus courtes.
3. *Neutralité*. Utilisez un style neutre. N'utilisez pas le « je », sauf dans les remerciements.
4. *Conjugaison*. La voix active est plus directe que la voix passive. Utilisez le présent autant que possible pour un style plus dynamique (sauf dans la conclusion où l'on peut utiliser le passé). Les phrases doivent contenir un verbe.
5. *Orthographe*. Vérifiez l'orthographe et les fautes grammaticales.
6. *Typographie*. Respectez les règles typographiques propres à la langue utilisée (e.g., types de guillemets, pas d'espace avant une virgule, espace avant un « : » ou un « ? » en français mais pas en anglais).
7. *Exemples*. Illustrez les concepts importants ou complexes par des exemples simples.
8. *Acronymes*. Evitez d'utiliser trop d'abréviations et acronymes si ce n'est pas nécessaire. Si vous utilisez une abréviation il faut d'abord l'introduire.


Pour améliorer son style, le livre de Strunk et White [14] est une excellente référence. Les consignes présentées ci-dessus s'appliquent également aux titres et *a fortiori* au titre du travail. Elles sont illustrées ci-dessous par quelques contre-exemples et exemples.


[Précision : formalisme et cohérence]

 Le nombre de nœuds d'un graphe est noté  $n$ . Un graphe est un ensemble de sommets  $S$  et d'arêtes entre nœuds. (...) Un graphe pour lequel  $|S| = 0$  est un graphe vide.

 Un *graphe non-dirigé*  $G$  est une paire ordonnée  $(S, A)$ , où  $S$  est un ensemble fini d'éléments appelés *sommets* et  $A$  est un ensemble de paires non-ordonnées de sommets distincts de  $S$ . Chaque élément  $\{s, t\} \in A$  est une *arête* reliant les sommets  $s$  et  $t$ . On note  $n$  le *nombre de sommets* de  $G$ . (...) Si  $n = 0$ , alors le graphe est *vide*.


[Précision : cohérence]


 (...) software evolution (...) software change (...) software modification (...)

 Ces notions sont-elles les mêmes ? Si seulement la première a été définie formellement dans le texte, le lecteur peut en douter. Evitez d'utiliser des synonymes pour une même notion.


7. Notez que le non-spécialiste est tout de même professeur ou étudiant en informatique. Ce n'est pas à votre grand-mère qu'il faut s'adresser !


*[Concision]*

 Pour sauvegarder des données on peut le faire sous forme de fichiers textes, qui représentent les données dans une forme lisible par l'être humain, par une séquence de caractères ; ou par des fichiers binaires dans lesquels les données sont représentées par des octets.


 Il y a deux manières différentes de stocker des données : dans un format *texte* ou *binaire*. Dans un format *texte*, les données sont représentées par une séquence de *caractères*. Ces données sont lisibles pour un être humain. Dans un format *binaire*, les données sont représentées par une séquence d'*octets*.


*[Neutralité]*

 Je commençais à regretter d'avoir voulu programmer les quatre opérations, mais je m'étais fixé un but et je voulais l'atteindre. De toute façon, cela rend le projet encore plus passionnant !


 Quel est l'intérêt de savoir cela ? Evitez ce genre de réflexion personnelle.

*[Concision, précision, neutralité]*


 Par la pratique, on remarqua que des couleurs pouvaient être obtenues en mélangeant d'autres couleurs. On est assez vite arrivé à quelques couleurs de base à partir desquelles on pouvait représenter toutes les couleurs. Ces fameuses couleurs sont appelées couleurs de base : le rouge, le vert et finalement le bleu. A partir de ces couleurs, il est donc possible par un savant mélange de représenter toutes les couleurs.

 Toute couleur peut s'obtenir par un mélange de trois couleurs de base : le rouge, le vert et le bleu.

*[Conjugaison]*


 L'algorithme sera exécuté en temps  $O(n)$ .


- Nous avons d'abord essayé la méthode  $X$  mais (...).
- La valeur 0 signifiant le noir et la valeur 1 représentant le blanc.


 La complexité en temps de l'algorithme est  $O(n)$ .

- La méthode  $X$  ne convient pas car (...).
- La valeur 0 représente le noir et la valeur 1 représente le blanc.

*[Acronymes, précision]*

 L'objet de ce travail est de développer un CMS.

 L'objet de ce travail est de développer un *Contents Management System (CMS)*. Un *CMS* est un système de gestion de contenu, c'est-à-dire (...).

 Quand vous avez terminé d'écrire les premières pages de votre rapport, retournez voir votre directeur. Les défauts de style sont souvent récurrents et les corriger rapidement fera gagner du temps, à vous comme à elle/lui.

## 4.4 Présentation

Pour bien concevoir la mise en page de votre travail, utilisez un bon logiciel de traitement de texte, comme  $\text{\LaTeX}$  (cf. Section 2.3). Choisissez les polices et la taille de caractères avec soin et évitez d'en changer trop. En général, on utilise un texte *justifié* pour les paragraphes. La hiérarchie des titres doit être cohérente visuellement (e.g., taille des titres).

Vous pouvez utiliser des notes de bas de page (mais pas trop) pour donner des précisions qui ne sont pas nécessaires à la compréhension du texte. Une note de bas de page ne sera pas toujours lue. Pour être sûr qu'une précision soit lue, on peut la donner dans le texte entre parenthèses.

D'autres éléments, présentés ci-dessous, facilitent la lecture et améliorent la présentation : environnements, références croisées, tableaux et figures.

### Environnements

Utilisez des environnements pour faire ressortir les éléments importants du texte.

**Définition 1.** Un *environnement* est une partie du texte qui a un rôle bien défini, comme un théorème, une définition, un exemple, une démonstration, un morceau de code. Cette partie du texte est formatée de telle sorte que l'on puisse identifier directement l'environnement et son rôle. Certains environnements sont numérotés pour que l'on puisse y faire référence ailleurs dans le texte.

**Exemple 5.** La Définition 1 est présentée dans un environnement utilisé pour les définitions importantes. Cet exemple est lui-même présenté dans un environnement dont le rôle est d'illustrer certaines notions.



L'Algorithme A a une complexité  $O(n^2)$  car (longs arguments...). Il donne le résultat souhaité car (longs arguments...).



**Proposition 1.** La complexité de l'Algorithme A est en  $O(n^2)$  où  $n$  représente le nombre d'éléments dans le tableau donné en entrée.

*Démonstration.* (longs arguments...) □

**Théorème 1** (Exactitude de l'Algorithme A). *Quand l'Algorithme A se termine, le tableau résultat contient les nombres triés par ordre croissant.*

*Démonstration.* (longs arguments...) □

### Références croisées

**Définition 2.** Tout élément du texte (section, équation, définition, référence bibliographique, figure, théorème, tableau, etc.) qui possède une *étiquette* (sous la forme d'un numéro ou de quelques lettres) peut être cité dans le texte en utilisant cette étiquette. On appelle cette citation une *référence croisée*.

**Exemple 6.** Nous avons introduit la notion d'environnement dans la Définition 1.

On utilise une majuscule pour faire une référence croisée vers un environnement numéroté, une figure, ou un tableau (« Voir Proposition x », « comme illustré dans le Tableau y »).

**Exemple 7.** Dans ce document, il n'y a pas beaucoup d'équations. Nous présentons cependant l'Equation (1) dans la formulation du Théorème de Green (cf. Théorème 2, p. 23).

## Figures et tableaux

Les tableaux et les figures facilitent la lecture et permettent d'illustrer des résultats ou des observations. Une figure peut aider la compréhension mais ne constitue pas un argument en soi. Il faut justifier les choix ou les résultats dans le texte. Un tableau permet de résumer des éléments. Il peut par exemple être utilisé dans la comparaison de différentes approches, pour donner les résultats de tests ou d'une étude empirique.



Une figure ou un tableau doivent *toujours* être numérotés et accompagnés d'une légende. En général, la légende d'une figure est placée en dessous de celle-ci. C'est le contraire pour un tableau : la légende se trouve au dessus.



N'utilisez pas de figures superflues qui ne sont pas référencées dans le texte. Une référence croisée dans le texte devrait apparaître pour chaque figure ou tableau.

## 5 Après la rédaction

La rédaction est un processus itératif. Chaque partie rédigée doit être révisée plusieurs fois, jusqu'à l'obtention du résultat souhaité. Il ne faut pas avoir peur de « jeter » une partie de texte superflue, de restructurer le texte, de le lire et le relire, en se mettant à la place du lecteur.

Quand vous avez une première version (presque) complète de votre travail, utilisez les moyens disponibles pour le réviser :

- utilisez *toujours* un vérificateur d'orthographe ;
- utilisez un dictionnaire et un *bescherelle* pour la grammaire et la conjugaison ;
- faites relire le travail à différentes personnes (entourage pour la clarté, l'orthographe et la grammaire ; directeur pour le contenu) ;
- révissez soigneusement l'introduction et la conclusion ;
- vérifiez l'homogénéité des notations et des termes utilisés ;
- améliorez l'aspect graphique du travail, regardez votre document « de loin » pour voir si la mise en page est esthétique : placement des figures, grand espace blanc au milieu d'une section, *etc.*



Soignez l'orthographe : lisez et relisez votre travail ; faites le relire par quelqu'un d'autre ; utilisez un correcteur orthographique.



Il est difficile d'avoir du recul sur son texte après l'avoir travaillé de nombreuses heures. Ne craignez pas la critique : les relecteurs vous rendent service et vous permettent de vous améliorer.

## 6 Présentation orale

Une fois le travail rédigé et rendu, vient la présentation orale. Les principes et les conseils donnés pour la rédaction sont de manière générale les mêmes pour un exposé :

- avoir un bon plan ;
- être clair et structuré ;
- savoir à qui l'on s'adresse, se mettre à la place de l'auditeur qui n'est pas spécialiste du sujet ;
- introduire les objectifs ;



- définir les notions la première fois qu’elles sont utilisées ;
- présenter les contributions personnelles ;
- conclure.

Il existe cependant des différences entre l’écrit et l’oral qui concernent le contenu et la forme.

- a) *Contenu*. Le rapport écrit doit être complet et précis. La durée d’une présentation orale est courte. Il est impossible de couvrir plusieurs dizaines de pages en quelques minutes : n’essayez pas de tout dire. Il faut aller à l’essentiel, donner les grandes idées mais sans rentrer dans des détails trop techniques. Si vous ne pouvez pas parler de tous vos chapitres, choisissez les plus représentatifs et citez simplement ce que vous avez fait dans les autres.

Si le contenu n’est pas identique au texte, le plan de l’exposé ne doit pas l’être non plus. N’essayez pas à tout prix de suivre le plan de votre travail écrit. Construisez un plan de l’exposé qui structure au mieux les idées importantes présentées.

- b) *Forme*. Une présentation est un exercice *oral*, dont les transparents ne sont qu’un support. Dans un rapport écrit, il est essentiel d’utiliser des phrases. Sur un transparent, il ne doit pas y avoir trop d’informations. Utilisez des mots clés, des illustrations ou des phrases courtes. Ne présentez qu’une seule ou peu d’idées dans un même transparent.

Comme dans le rapport écrit, il est important d’annoncer les résultats *au début* de l’exposé (« Dire ce que l’on va dire avant de le dire » [3]). Les auditeurs voudront en savoir plus et seront intéressés par la suite. Il est également important de conclure en rappelant les résultats. Cela permet au public de *retenir* le message.

Pour clarifier les concepts abstraits, utilisez des exemples concrets, simples mais intéressants. Cela permet au public de suivre et évite à l’orateur de longues explications.



Ne récitez pas le texte des transparents. Le public sait lire et désire que vous expliquiez votre travail : les transparents ne sont qu’un support. Regardez votre auditoire. C’est au public que vous vous adressez et non à l’écran.



Prenez le temps d’expliquer chaque transparent. Cela implique qu’il n’y en ait pas trop et que ceux-ci ne soient pas trop denses. Comptez environ une à deux minutes par transparent. Exercez-vous et chronométrez le temps de votre présentation. Si vous prenez trop de temps, révisez vos transparents plutôt que d’accélérer le rythme.

## 7 Conclusion

Un texte scientifique et technique permet à son auteur de présenter son travail. Il met en avant les résultats et les contributions personnelles. Il est important que celui-ci soit de bonne qualité. Les éléments fondamentaux de la rédaction ont été présentés. Le plan de rédaction est structuré et logique, comme la structure globale et interne des sections et des paragraphes. Le contenu est rigoureux et présenté de manière progressive (les grandes idées avant de les détailler). Le style est précis, concis, actif, dynamique et neutre. La présentation est claire et esthétique. La rédaction scientifique doit s’inclure dans une démarche scientifique. Il convient de mettre en avant ce qui est personnel et de citer les travaux sur lesquels on se base. L’utilisation des mots d’un autre auteur se fait en suivant des conventions simples mais précises, pour éviter le plagiat.

La méthode de rédaction proposée dans cette note comporte deux grandes étapes : l’organisation des idées et la rédaction en elle-même. Chacune de ces étapes est itérative : le plan de rédaction et le texte doivent être révisés jusqu’à l’obtention d’un résultat qui exprime au mieux les idées.

Quelques conseils ont également été donnés pour la présentation orale d’un travail scientifique et technique.

Cette note contient de nombreux exemples et conseils permettant au lecteur d'améliorer sa rédaction et sa méthode de travail. Les exemples ont été choisis dans le domaine de l'informatique. Ce document constitue donc une aide pratique pour les étudiants en informatique – dans le cadre de leurs travaux (*e.g.*, rapports, mémoire) – et pour leurs enseignants qui ne devront plus répéter les mêmes consignes.

## Remerciements

Je tiens à remercier mes collègues V. Bruyère, A. Buys, J. Cardinal (ULB), A. Decan, O. Delgrange, T. Mens et J. Wijzen pour leurs suggestions. Beaucoup de précieux conseils de rédaction m'ont été donnés par P. Hansen lors de l'écriture de nos articles communs. Je l'en remercie sincèrement, cela m'a été très utile pour écrire cette note. Je remercie également certains étudiants – dont je tairai les noms – pour leur participation involontaire dans le choix de « contre-exemples » illustratifs.

## Références

- [1] AHO, A., AND ULLMAN, J. *Concepts fondamentaux de l'informatique*. Dunod, Paris, 1996.
- [2] BIBLIOTHÈQUES DE L'ULB. Eviter le plagiat. URL : [www.bib.ulb.ac.be/fr/aide/eviter-le-plagiat/](http://www.bib.ulb.ac.be/fr/aide/eviter-le-plagiat/), (consulté le 3/10/2007).
- [3] BRUYÈRE, V. Comment bien rédiger. Exposé à l'intention des étudiants de 1ère licence en informatique, Université de Mons-Hainaut, 2006.
- [4] CORMEN, T.H., LEISERSON, CH.E., RIVEST, R.L., AND STEIN, C. *Introduction to algorithms*, 2nd ed. The MIT Press, London, 2002.
- [5] GOOSSENS, M., MITTELBAACH, F., AND SAMARIN, A. *The LaTeX Companion*. Addison-Wesley, Boston, 2004.
- [6] HORSTMANN, C. *Java Concepts*, 4th ed. Wiley, New Jersey, 2006.
- [7] LATEX. A Document Preparation System. URL : [www.latex-projet.org](http://www.latex-projet.org), Version : LaTeX2ε (consulté le 14/9/2007).
- [8] LINDVALL, M., AND SANDAHL, K. How Well do Experienced Software Developers Predict Software Change ? *J. Systems and Software* 43, 1 (1998), 19–27.
- [9] LYX. The Document Processor. URL : [www.lyx.org](http://www.lyx.org), Version : 1.5.1 (consulté le 14/9/2007).
- [10] NADJI, F., AND BOUDIA, D. Guide de gestion des références bibliographiques. URL : [docinsa.insa-lyon.fr/refbibli/contenus/doc/guide\\_ref\\_bib\\_2004.pdf](http://docinsa.insa-lyon.fr/refbibli/contenus/doc/guide_ref_bib_2004.pdf), INSA, Lyon, 2004 (consulté le 21/9/2007).
- [11] OETIKER, T., PARTL, H., HYNÄ, I., AND SCHLEGL, E. The Not So Short Introduction to L<sup>A</sup>T<sub>E</sub>X2 $\epsilon$ . URL : [ctan.tug.org/tex-archive/info/lshort/english/](http://ctan.tug.org/tex-archive/info/lshort/english/), 2007 (consulté le 14/9/2007).
- [12] OETIKER, T., PARTL, H., HYNÄ, I., AND SCHLEGL, E. Une courte (?) introduction à L<sup>A</sup>T<sub>E</sub>X2 $\epsilon$ . Traduit par HERRB, M.. URL : [ctan.tug.org/tex-archive/info/lshort/french/](http://ctan.tug.org/tex-archive/info/lshort/french/), 2001 (consulté le 14/9/2007).
- [13] PALME, J. Plagiarism. URL : [people.dsv.su.se/~jpalme/plagiarism.html](http://people.dsv.su.se/~jpalme/plagiarism.html), (consulté le 20/9/2007).
- [14] STRUNK, W. JR., AND WHITE, E.B. *The Elements of Style*, 4th ed. Macmillan, New York, 1979.
- [15] VALDURIEZ, P. Some Hints to Improve Writing of Technical Papers. *Ingénierie des Systèmes d'Informations* 2, 3 (1994), 371–375.

## Annexe A : Abréviations latines courantes

Dans la lecture de textes en anglais ou en français, vous rencontrerez souvent des abréviations latines. Leur utilisation est très fréquente dans les articles scientifiques. Le Tableau 3 donne les significations des abréviations les plus courantes, ainsi que des alternatives francophones ou symboliques. Vous pouvez bien sûr vous passer des abréviations et écrire les mots en toutes lettres.

TABLE 3 – Significations des abréviations latines courantes

Abréviation	Mots latins	Signification	Alternative
<i>e.g.</i>	<i>exempli gratia</i>	par exemple	par ex.
<i>i.e.</i>	<i>id est</i>	c'est-à-dire	c.-à-d.
<i>cf.</i>	<i>confer</i>	consulter	voir
<i>etc.</i>	<i>et cetera</i>	et le reste	...
<i>et al.</i>	<i>et alia</i>	et autres	
<i>Q.E.D.</i>	<i>quod erat demonstrandum</i>	ce qu'il fallait démontrer	C.Q.F.D. ou □

### Remarques :

1. Dans votre texte, choisissez ce que vous préférez mais restez cohérents (évitez de mélanger sans cesse « *i.e.* » et « c.-à-d. » par exemple).
2. Respectez la typographie exacte des abréviations (les points sont importants puisqu'ils marquent le fait que c'est une abréviation).
3. Les abréviations latines sont souvent écrites en italique, comme les mots latins qui ne sont pas abrégés (par exemple *a priori* ou *a fortiori*).
4. On confond souvent « *e.g.* » et « *i.e.* » car ces abréviations servent toutes les deux à clarifier ce qui précède. Pourtant, leurs significations (« par exemple » et « c'est-à-dire ») sont différentes. La première est utilisée pour illustrer à l'aide d'un exemple, la seconde permet de reformuler une idée différemment ou de définir une notion.
5. Les abréviations « *e.g.* » et « *i.e.* » sont suivies d'une virgule.
6. Il ne faut pas mettre de points de suspension après « *etc.* » (redondance).
7. Une liste d'exemples qui commence par « *e.g.* » ne doit pas se terminer par « *etc.* » ou par des points de suspension (redondance).
8. L'abréviation « *et al.* » est souvent utilisée quand on cite une référence contenant un grand nombre d'auteurs (par exemple : Cormen *et al.* [4]).
9. Ecrire « et *etc.* » est redondant.

## Annexe B : Quelques conseils L<sup>A</sup>T<sub>E</sub>X

Nous présentons ici quelques exemples et conseils relatifs au système L<sup>A</sup>T<sub>E</sub>X. Pour chacun de ceux-ci, la section de ce document à laquelle il se réfère est indiquée. Il ne s'agit pas d'une documentation complète, mais de quelques illustrations et astuces. Pour plus d'informations, il faut se référer à la documentation donnée dans la Section 2.3.

**Remarque 5.** Dans les conseils qui suivent, il est parfois suggéré d'inclure un package. Pour ce faire, il suffit d'ajouter la commande `\usepackage{nomPackage}` dans le préambule du document (i.e., entre les commandes `\documentclass` et `\begin{document}`).

**Editer facilement du contenu scientifique (cf. Section 2.3, p. 7)**

Dans l'exemple ci-dessous, le théorème et l'équation sont formatés et numérotés automatiquement. On entoure les notations mathématiques avec le signe \$.

CODE L<sup>A</sup>T<sub>E</sub>X

---

```
\begin{thm}[Green] Soit $C$, une courbe plane simple, positivement
orientée et $C^1$ par morceaux, $D$ le domaine compact lisse du plan
délimité par $C$ et $Pdx + Qdy$ une 1-forme différentielle sur
$\mathcal{R}^2$. Si $P$ et $Q$ ont des dérivées partielles continues
sur une région ouverte incluant $D$, alors :
\begin{equation}
\int_C P dx + Q dy = \iint_D \frac{\partial Q}{\partial x} - \frac{\partial P}{\partial y} dx dy
\end{equation}
\end{thm}
```

RÉSULTAT

---

**Théorème 2** (Green). *Soit \$C\$, une courbe plane simple, positivement orientée et \$C^1\$ par morceaux, \$D\$ le domaine compact lisse du plan délimité par \$C\$ et \$Pdx + Qdy\$ une 1-forme différentielle sur \$\mathbb{R}^2\$. Si \$P\$ et \$Q\$ ont des dérivées partielles continues sur une région ouverte incluant \$D\$, alors :*

$$\int_C Pdx + Qdy = \iint_D \frac{\partial Q}{\partial x} - \frac{\partial P}{\partial y} dx dy \quad (1)$$


---

**Gérer les références bibliographiques (cf. Section 3.2, p. 9)**

L'outil BibTeX [5] est disponible avec la plupart des distributions L<sup>A</sup>T<sub>E</sub>X. Avec celui-ci, les références bibliographiques sont mises en page automatiquement selon le format choisi, à partir d'un fichier .bib contenant les entrées bibliographiques. Voici un exemple d'entrée BibTeX :

```
@book{Goossens04,
  author = {Goossens, M. and Mittelbach, F. and Samarin, A.},
  title = {{The LaTeX Companion}},
  publisher = {Addison-Wesley},
  address = {Boston},
  year = {2004}
}
```

On peut citer cette référence dans le fichier .tex comme suit : `\cite{Goossens04}`. On trouve facilement des entrées BibTeX pour les publications en informatique sur le web (e.g., le site *DBLP*, [www.informatik.uni-trier.de/~ley/db/index.html](http://www.informatik.uni-trier.de/~ley/db/index.html)).

Avec cet outil, on peut changer très facilement la manière dont sont créées les étiquettes : numéro ou noms des auteurs et date, etc. Un autre avantage est que seules les références réellement utilisées dans le texte apparaissent dans la bibliographie.

### Références bibliographiques groupées (cf. Section 3.2, p. 9)

Le package `cite` permet d'ordonner plusieurs références citées en même temps et de les afficher sous la forme d'un intervalle si elles se suivent.

CODE  $\LaTeX$  \_\_\_\_\_

```
Voici quelques références groupées~\cite{lyx, latexweb, Strunk79, Goossens04, Lindvall98}.
```

RÉSULTAT SANS LE PACKAGE `cite` \_\_\_\_\_

Voici quelques références groupées [9, 7, 14, 5, 8].

RÉSULTAT AVEC LE PACKAGE `cite` \_\_\_\_\_

Voici quelques références groupées [5, 7–9, 14].

---

### Faire une citation exacte (cf. Section 3.3, p. 10)

Pour faire une citation exacte en décalant un paragraphe, on utilise l'environnement `quote`.

CODE  $\LaTeX$  \_\_\_\_\_

```
En ce qui concerne les illustrations, Valduriez précise qu'il peut s'agir de dessins,
d'exemples ou d'algorithmes et donne le conseil suivant~\cite[p. 374]{Valduriez94}:
\begin{quote}
In all cases, avoid illustrations that are either too simple, \emph{e.g.}, a 2-line
algorithm, or two complex, \emph{e.g.}, a 50-line algorithm. Like text, drawings
should be brief and precise, avoiding unnecessary coloring and details.
\end{quote}
```

RÉSULTAT \_\_\_\_\_

En ce qui concerne les illustrations, Valduriez précise qu'il peut s'agir de dessins, d'exemples ou d'algorithmes et donne le conseil suivant [15, p. 374] :

In all cases, avoid illustrations that are either too simple, *e.g.*, a 2-line algorithm, or two complex, *e.g.*, a 50-line algorithm. Like text, drawings should be brief and precise, avoiding unnecessary coloring and details.

---

### Structure et table des matières (cf. Section 4.1, p. 11)

Les différentes sections sont définies dans le texte avec les commandes `\chapter`, `\section`, `\subsection` et `\subsubsection`. Les titres sont alors mis en forme et numérotés par  $\LaTeX$ . Pour insérer une table des matières dans le texte, similaire à celle de la première page de ce document, on utilise simplement la commande `\tableofcontents`.

**Remarque 6.** *Il faut compiler deux fois le fichier `.tex` pour que cette table des matières soit correcte. La première compilation écrit les noms des sections et leurs numéros de pages dans un fichier `.toc`. La seconde compilation utilise ce fichier pour mettre à jour la table des matières.*

*Dans certains cas (quand la table fait plus d'un page), une troisième compilation peut être nécessaire.  $\LaTeX$  le précise dans les messages affichés par le compilateur.*

**Ecrire des algorithmes (cf. Section 4.2, p. 15)**

Pour écrire un algorithme en pseudo-code, on peut utiliser les packages `algorithm` et `algorithmic`.

**CODE L<sup>A</sup>T<sub>E</sub>X**


---

```

\begin{algorithm}
\caption{Recherche linéaire du maximum}
\begin{algorithmic}[1]
\REQUIRE un tableau d'entiers $A$
\ENSURE la valeur du plus grand entier contenu dans $A$
\STATE $\max \leftarrow -\infty$
\FOR{$i \leftarrow 1$ à $longueur[A]$}
\IF{$\max < A[i]$}
\STATE $\max \leftarrow A[i]$
\ENDIF
\ENDFOR
\RETURN $\max$
\end{algorithmic}
\end{algorithm}

```

**RÉSULTAT****Algorithme 1** Recherche linéaire du maximum

**Entrée:** un tableau d'entiers  $A$

**Sortie:** la valeur du plus grand entier contenu dans  $A$

- 1:  $max \leftarrow -\infty$
  - 2: **pour**  $i \leftarrow 1$  à  $longueur[A]$  **faire**
  - 3:     **si**  $max < A[i]$  **alors**
  - 4:          $max \leftarrow A[i]$
  - 5:     **fin si**
  - 6: **fin pour**
  - 7: **retourner**  $max$
- 

Notez que pour produire le résultat ci-dessus, on a redéfini les noms des commandes dans le préambule :

```

\floatname{algorithm}{Algorithme}
\renewcommand{\algorithmicrequire}{\textbf{Entrée:}}
\renewcommand{\algorithmicensure}{\textbf{Sortie:}}
\renewcommand{\algorithmicif}{\textbf{si}}
\renewcommand{\algorithmicthen}{\textbf{alors}}
\renewcommand{\algorithmicelse}{\textbf{sinon}}
...

```

**Typographie (cf. Section 4.3, p. 16)**

Le package `babel` permet d'appliquer automatiquement certaines règles typographiques propres à une langue. Les « guillemets français » s'obtiennent avec les commandes `\og` (ouvrez les guillemets) et `\fg` (fermez les guillemets).

CODE L<sup>A</sup>T<sub>E</sub>X

---

```
\documentclass{article}
\usepackage[french,frenchb]{babel}
\begin{document}
Voici un exemple: voyez-vous comment sont \og ajoutés \fg\ certains espaces?
\end{document}
```

## RÉSULTAT

---

Voici un exemple : voyez-vous comment sont « ajoutés » certains espaces ?

---

**Notes de bas de page (cf. Section 4.4, p. 18)**

Définir une note de bas de page se fait avec la commande `\footnote`.

CODE L<sup>A</sup>T<sub>E</sub>X

---

```
Alan Turing\footnote{23 Juin 1912 -- 7 Juin 1954.} est à l'origine des
ordinateurs tels que nous les connaissons.
```

## RÉSULTAT

---

Alan Turing<sup>8</sup> est à l'origine des ordinateurs tels que nous les connaissons.

---

**Environnements (cf. Section 4.4, p. 18)**

Il est facile de définir vos propres environnements en L<sup>A</sup>T<sub>E</sub>X (voir par exemple [11]). Ils seront numérotés et formatés automatiquement. Les packages `amsmath` et `amsthm` sont très utiles pour plusieurs raisons. L'une d'elle est qu'ils permettent d'obtenir une série d'environnements prédéfinis comme `proof` qui ajoute un carré (□) pour marquer la fin d'une démonstration (remplace le C.Q.F.D. un peu « scolaire »). Notez que le mot *proof* est traduit automatiquement en *démonstration* grâce à l'inclusion du package `[french,frenchb]{babel}`.

CODE L<sup>A</sup>T<sub>E</sub>X

---

```
\begin{proof}
Par induction. (\ldots)
\end{proof}
```

## RÉSULTAT

---

*Démonstration.* Par induction. (...) □

---



---

8. 23 Juin 1912 – 7 Juin 1954.

**Références croisées (cf. Section 4.4, p. 18)**

Les références croisées sont gérées automatiquement par  $\LaTeX$ . Cela signifie que si on ajoute une définition avant la Définition 1, l'étiquette de celle-ci et des suivantes sont automatiquement incrémentées dans le texte. Pour ce faire, on définit un `label` et on crée une référence croisée en utilisant la commande `ref`. On peut également obtenir un numéro de page en utilisant la commande `pageref`.

**CODE  $\LaTeX$** 


---

```
\begin{defn} \label{def_algo}
Un \emph{algorithme} est une séquence finie d'étapes non-ambiguës
permettant de résoudre un problème.
\end{defn}
Dans la Définition \ref{def_algo} (\emph{cf.} p. \pageref{def_algo}), un élément
important est manquant : la notion d'entrées et sorties d'un algorithme.
```

**RÉSULTAT**


---

**Définition 3.** Un *algorithme* est une séquence finie d'étapes non-ambiguës permettant de résoudre un problème.

Dans la Définition 3 (cf. p. 27), un élément important est manquant : la notion d'entrées et sorties d'un algorithme.

---

**Remarque 7.** Il faut compiler deux fois le fichier `.tex` pour que  $\LaTeX$  puisse gérer les références croisées. La première compilation écrit les étiquettes et leurs numéros de pages dans un fichier `.aux`. La seconde compilation utilise ce fichier pour mettre à jour les références croisées.

**Correcteur orthographique (cf. Section 5, p. 19)**

Il existe plusieurs vérificateurs d'orthographe qui peuvent être utilisés sur vos fichiers `.tex`. Tout dépend de votre installation (système d'exploitation, éditeur utilisé). Par exemple, *TeXShop* (Mac OS X) inclus un vérificateur. Sous *Linux*, on peut utiliser la commande `ispell`.

**Editer des transparents en  $\LaTeX$  (cf. Section 6, p. 19)**

Si vous avez écrit votre travail avec l'aide de  $\LaTeX$ , pourquoi ne pas l'utiliser pour les transparents lors de votre présentation ? Il existe plusieurs packages et classes pour le faire, comme *Beamer* et *Prosper*. Ils permettent de faire des présentations dynamiques (transparents incrémentaux pour « animer » une figure ou pour dévoiler du texte progressivement).

Vous devrez installer ces packages car ils ne font pas partie de la distribution standard de  $\LaTeX$ . Ils sont disponibles aux adresses suivantes :

- *Beamer* : [sourceforge.net/projects/latex-beamer/](http://sourceforge.net/projects/latex-beamer/),
- *Prosper* : [sourceforge.net/projects/prosper/](http://sourceforge.net/projects/prosper/).