

# Guidelines pour la réalisation de projets et la rédaction de rapports

Version : 4 avril 2012

---

## Généralités

Ce document énonce les diverses exigences que doivent rencontrer les projets que vous réaliserez pour les cours du Pr. Louis Wehenkel. Ces exigences doivent être respectées. Leur non-respect pourra entraîner une diminution de la note attribuée. Vous êtes donc invités à lire ce document attentivement.

## Guidelines pour le rapport

1. Votre rapport doit comporter vos nom, prénom, matricule et section.
2. Chaque énoncé mentionne un *nombre de pages conseillé*. Celui-ci permet de donner une indication à l'étudiant afin qu'il ne remplisse pas le rapport avec des éléments inutiles. Ce nombre de pages conseillé s'entend sans les codes sources qui seront placés en annexe. Cette limite n'est pas stricte. Remarquez cependant, et cela s'applique à tous les rapports que vous serez amenés à rédiger au cours de votre vie, qu'un rapport inutilement long fatigue le lecteur et peut diminuer l'attention, l'intérêt ou la bienveillance qu'il porte au document que vous avez rédigé.
3. Il n'est pas nécessaire d'écrire une introduction, des rappels des questions posées ou de faire des rappels théoriques (sauf si cela est explicitement demandé). De même, il n'est pas nécessaire, si le rapport n'excède pas 20 pages, d'introduire une table des matières.
4. Toute question ou sous-question posée dans l'énoncé devra comporter un élément de réponse dans le rapport.  
*Exemple* : si il existe, dans l'énoncé, un point numéroté (2 b), le rapport devra comporter une partie (2 b) contenant les éléments de réponse.
5. Pour toute question précédée du symbole , il est demandé de rendre, dans l'archive que vous soumettrez, les fichiers source contenant le code que vous avez rédigé afin de répondre à la question ainsi que tous les fichiers nécessaires à l'exécution du code. Ces codes devront, de plus, être ajoutés en annexe du rapport. Il n'est pas nécessaire de rendre de code (ni dans l'archive, ni dans les annexes) pour les questions non-précédées de ce symbole.

6. N'introduisez pas vos codes source dans le corps du rapport. Le code source doit être ajouté à la fin du rapport sous forme d'annexe. Les codes devront être clairement séparés et lisibles.

*Exemple* : si le point (3 a) de l'énoncé vous demande de coder une fonction `f1`, ce point (3 a) du rapport ne devra pas contenir de ligne de code. Une phrase du type « *Le code source de f1 peut être trouvé dans le fichier f1.m et à l'annexe 3.* » est suffisante comme réponse (en plus de la justification détaillant la manière dont vous avez codé la fonction). Si une autre question vous est également posée à ce point, vous devez cependant y répondre.

7. Justifiez toutes les approches et méthodes utilisées ainsi que vos raisonnements.
8. Toutes les figures et tous les tableaux doivent porter une référence claire. Aucune ambiguïté ne doit être possible entre les différentes figures et tableaux. De plus, toute réponse qui fait référence à une figure ou à un tableau doit faire explicitement référence à cette figure ou ce tableau.

*Exemple* : « Comme nous pouvons le constater à la Figure 1, ... ».

9. Tout graphique doit comporter un titre, une légende ainsi que des noms et des échelles pour les axes. Vous devez, de plus, faire en sorte que le graphique soit lisible (choix des échelles de manière adéquate, épaisseur des traits, couleurs, etc.).
10. Votre rapport devra être rendu au format pdf.
11. Respectez les consignes et spécifications de l'énoncé.

*Exemple* : si vous devez implémenter une fonction, les entrées et sorties doivent suivre strictement les instructions données.

*Exemple 2* : vous devez rendre votre projet dans les délais impartis.

12. Les réponses que vous rédigez doivent être structurées et lisibles :
  - utilisation correcte de la ponctuation ;
  - correction de l'orthographe, de la syntaxe et de la grammaire.

Il est important que le lecteur puisse aisément suivre votre raisonnement. Utilisez des paragraphes pour segmenter celui-ci et ainsi faciliter la lecture. Il est également important de ne pas polluer le texte principal avec des digressions inutiles qui déconcentrent le lecteur.

## Guidelines pour la programmation

1. Vous devez impérativement rendre vos fichiers source en plus du rapport. L'ajout des codes en annexe du rapport ne contredit en rien cette affirmation.
2. Les différents fichiers que vous rendez devront être clairement identifiables.

*Exemple* : si le point (1 c) demande la manipulation de code source, un fichier nommé `1c.m` ou `Q1c.m` devra être rendu.
3. Vos codes devront être lisibles et aisément compréhensibles. Appliquer quelques règles simples est suffisant :
  - indentez votre code ;

- espacez-le suffisamment ;
- commentez quand cela est nécessaire ;
- donnez un nom explicite aux variables et fonctions.

*Exemple* : référez-vous aux fichiers fournis avec les énoncés afin de vous faire une idée sur la manière de présenter votre code.

4. Les classes et fonctions que vous utiliserez devront être clairement documentées (descriptif de la fonction ou de la classe, description des arguments de la fonction, description des valeurs de sortie, etc.) de sorte que leur compréhension soit aisée.

*Exemple* : si vous rédigez une fonction, donnez-lui un nom explicite et indiquez clairement ce qu'elle fait, ce qu'elle prend comme arguments et les valeurs qu'elle renvoie. Cela ne s'applique qu'aux fonctions annexes que vous pourriez rédiger : si il vous est demandé au point (2 c) de l'énoncé de rédiger une fonction qui prend telle valeur en entrée et qui renvoie telle réponse, une fonction nommée Q2c fera parfaitement l'affaire.

5. Il vous appartient de vérifier que les codes que vous rendez fonctionnent directement et sans modification aucune sur les machines ms800 de l'Institut Montefiore sur lesquelles les codes seront testés.
6. Afin de répondre aux questions qui vous sont posées, vous pouvez utiliser toutes les toolboxes de Matlab disponibles sur les machines ms800 ainsi que les différents codes qui vous sont fournis avec l'énoncé.
7. Tous les codes rendus doivent être personnels, c'est-à-dire rédigés par vous-même. Vous pouvez néanmoins utiliser des fonctions écrites par d'autres personnes (amis ou Internet) et reprises avec leur consentement. Le fait que le code appartienne à une autre personne doit être clairement indiqué, en plus du nom de l'auteur, afin d'éviter tout plagiat. Notez cependant que les fonctions reprises ne doivent constituer qu'une aide à la résolution des questions qui vous sont posées et qu'elles ne doivent pas répondre directement ou indirectement à ce qu'il vous est demandé d'implémenter. Si vous avez le moindre doute, réimplémentez la fonction vous-même. Les correcteurs seront seuls juges en la matière. De plus, tout code réutilisé doit se conformer aux points 3 et 4 des Guidelines pour la programmation.

*Exemple* : si il vous est demandé d'implémenter une fonction qui effectue un traitement sur une matrice et qui renvoie la réponse sous forme d'une colonne et si vous trouvez une fonction toute faite sur Internet qui renvoie la solution sous forme d'une ligne. Utiliser cette fonction et transposer le résultat n'est pas autorisé, parce que la partie la plus importante du travail qui vous était demandé n'a pas été faite par vous.

*Exemple 2* : si, lors de l'implémentation de votre solution, vous souhaitez créer une structure de données de type `liste liée` et si vous vous procurez du code implémentant cette structure pour vous, alors vous avez le droit de l'utiliser à condition de bien spécifier que le code n'a pas été rédigé par vous et de mentionner le nom de l'auteur et le site Internet dont le code est issu.