

	ANNEE UNIVERSITAIRE 2014/2015 EXAMEN TP MACHINE Mention : Physique et Ingénierie UE : PY3002 Epreuve : Informatique pour la Physique Date : 16 décembre 2014, 14H durée : 3 heures Documents de cours et TP autorisés	Collège Sciences et technologies
---	---	--

Très important I

Logger vous sur votre compte personnel

- 1- Dans votre répertoire PY3002 , créer un répertoire examen_TP_2014
- 2- Ouvrir un fichier que vous nommerez fonctions.c
- 3- A la création de chaque nouveau fichier, insérer en ligne de commentaire le nom du fichier, votre prénom, votre nom et votre groupe, votre adresse mail, comme dans l'exemple suivant

```
/*fonctions.c */
/* TURING Alan / groupe A8 */
/*le 16/12/2014 */
/*alan.turing@etu.u-bordeaux.fr */
```

Les fichiers sans ce commentaire en en-tête seront considérés comme anonymes et ne seront pas corrigés.

Très important II

à la fin du DS, pour déposer vos fichiers, aller sur :

http://francois.marc.emi.u-bordeaux1.fr/examen_TP/

et suivre les consignes. Vous devez déposer vos **quatre** fichiers sources. Il est conseillé qu'ils s'exécutent avec les deux fichiers de test fournis. Vous pouvez aussi déposer vos fichiers SVG

Très important III

La qualité d'écriture du code et sa lisibilité seront largement pris en compte dans la notation.

Barème approximatif : I : 5 points ; II : 9 points ; III : 6 points (+ 2 points bonus)

Calcul des caractéristiques optiques d'un système de lentilles

Le but de cet exercice est d'écrire un programme permettant d'évaluer les caractéristiques d'un système optique (type lunette ou microscope) constitué de plusieurs lentilles dont la description est stockée dans un fichier, et de dessiner ce système.

Rappel et compléments sur les lentilles optiques :

Un système optique tel qu'une lunette astronomique ou un microscope optique est constitué d'une succession de lentilles. Pour une lentille mince, pour les petits angles d'incidence des rayons par rapport à l'axe de symétrie de la lentille, une construction géométrique (figure 1) et des formules simples (vues au semestre 1) donnent la relation entre la position et la taille de l'objet observé et celles de son image en fonction de la distance focale d_{focale} .

On notera que toutes les longueurs sont définies algébriquement.

Une distance focale positive correspond à une lentille convergente, une distance focale négative correspond à une lentille divergente.

Une distance objet (respectivement image) **positive** correspond à un objet (respect. image) **réel**. Une distance (objet ou image) **négative** correspond à un objet ou une image **virtuel(e)** (un lieu où les rayons convergeraient si on prolongeait les droites).

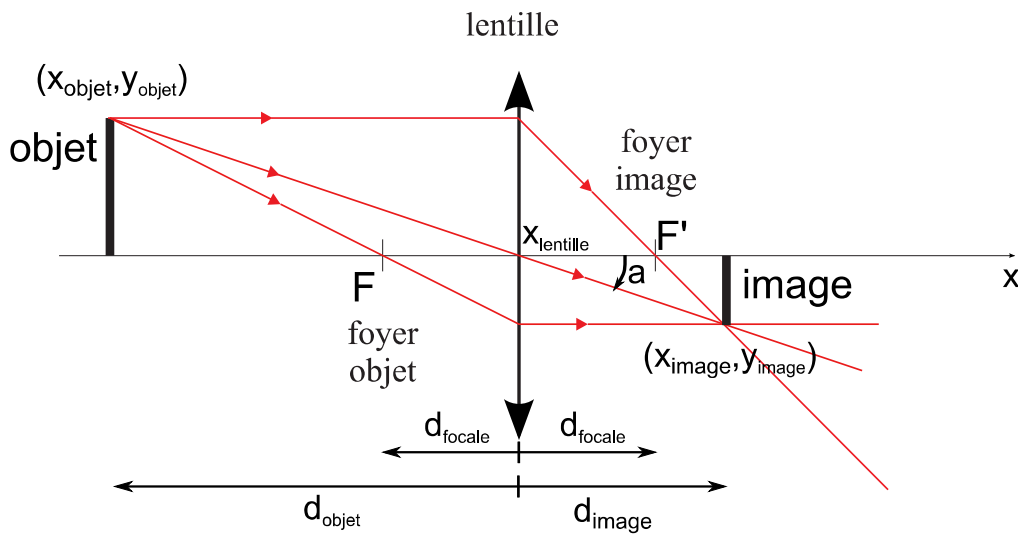


Figure 1 : Construction géométrique de l'image d'un objet. On notera que les distances (focale, objet et image) sont des longueurs algébriques qui sont toutes positives sur le schéma. L'angle α est négatif sur la figure.

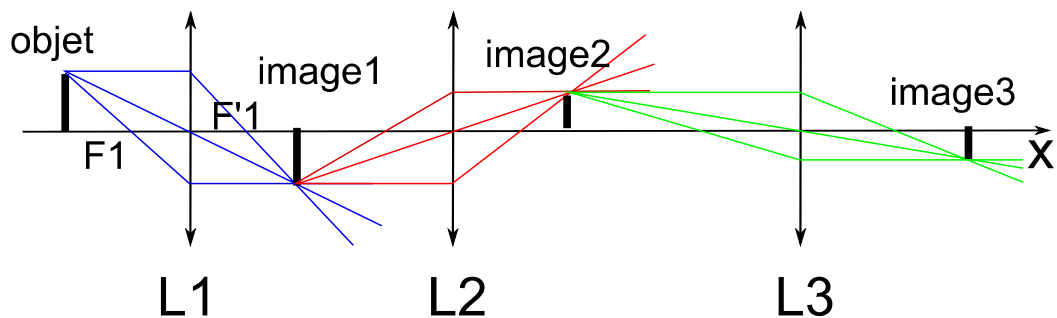


Figure 2 : Système optique à 3 lentilles (L1, L2, L3) : construction des images successives. Seuls les rayons servant à la construction ont été représentés, pas les rayons réels (qui traversent toutes les lentilles sans changer de direction entre les lentilles)

Relations (toutes les relations sont algébriques) entre les lieux remarquables

1. distances

$$\frac{1}{d_{\text{objet}}} + \frac{1}{d_{\text{image}}} = \frac{1}{d_{\text{focale}}} \quad (1)$$

2. angles et tailles objet et image

$$\tan \alpha = \frac{-y_{\text{objet}}}{d_{\text{objet}}} = \frac{y_{\text{image}}}{d_{\text{image}}} \quad (2)$$

I. ECRITURE DE FONCTIONS ET LENTILLE SIMPLE

Au cours de cette partie, les fonctions qui serviront par la suite au calcul des caractéristiques du système de lentilles seront écrites puis testées, avant d'être utilisées pour l'étude de la deuxième partie.

- I.1. Ecrire l'entête du programme avec les commentaires (nom, prénom, groupe, date).
Préparer les directives du préprocesseur ainsi que la fonction principale *main()*.
Enregistrer votre programme sous **fonctions.c**.

I.2. Ecriture des fonctions de bases

I.2.1. Ecrire la fonction *distance_image()* qui

- prend en entrées la distance objet et la distance focale
- renvoie la distance de l'image.

I.2.2. Ecrire la fonction *tangente_angle_sortie()* qui

- prend en entrées x_{objet} , y_{objet} et x_{lentille} ,
- renvoie la tangente de l'angle α de sortie du rayon passant par le centre de la lentille.

I.2.3. Tester ces deux fonctions depuis le programme principal, en prenant des valeurs correspondant au schéma de la figure 1. On pourra par exemple poser $x_{\text{lentille}} = 0,1\text{m}$ et $d_{\text{focale}} = 0,2\text{m}$. La valeur renvoyée par les fonctions doit être affichée à l'écran comme suit (valeurs arbitraires indiquées) :

Distance image : 0.043521 m

Tangente de l'angle : 0.043521

I.3. Ecriture des fonctions pour les ordonnées

I.3.2. Ecrire la fonction *position_image()*

- prend en entrées les abscisses et ordonnées x_{objet} , y_{objet} et x_{lentille} , et la distance focale
- calcule l'abscisse x_{image} et l'ordonnée y_{image} de l'image et fournit ces valeurs en sortie par un passage par adresse.
- ***Il est recommandé d'utiliser les fonctions précédentes***

I.3.3. Tester cette fonction depuis le programme principal, en prenant des valeurs correspondant au schéma de la figure 1. La valeur renvoyée par les fonctions doit être affichée à l'écran comme suit (valeurs arbitraires indiquées) :

Abscisse de l'image : 0.312345 m

Taille de l'image : 0.023545 m

II. CALCULS SUR DES SYSTEMES DE LENTILLES

On a défini les caractéristiques (simplifiées) des lentilles de systèmes optiques simples : une lunette astronomique, une lunette terrestre, un microscope. La description de ces systèmes est donnée dans des fichiers (les télécharger dans <http://francois.marc.emi.u-bordeaux1.fr/>) structurés comme suit :

- la première ligne contient le nombre **nb_lentilles** de lentilles,
- chacune des lignes suivantes contient les deux caractéristiques de la lentille : son abscisse **x_lentille** et sa distance focale **d_focale**.

Les lentilles sont données dans l'ordre où elles sont traversées par la lumière (donc l'objectif en premier), dans le sens croissant des abscisses..

II.1..Etape 1

Enregistrer le programme précédent sous **lentilles.c**.et modifier l'en-tête en conséquence.

II.2. Lecture des données d'entrées

Le programme doit recevoir comme informations en entrée :

- le nom du fichier à lire
- l'abscisse de l'objet *x_objet*
- la taille (=l'ordonnée) de l'objet *y_objet*

AU CHOIX :

II.2.a : Arguments par la ligne de commande (recommandé pour les tests, barème : +1)

Modifier le programme pour que celui-ci :

- a- lise les informations comme arguments de la ligne de commande.
- b- vérifie qu'il y a le bon nombre d'arguments,
- c- qu'il affiche si nécessaire, un message d'erreur et la syntaxe d'utilisation puis arrête le programme.

II.2.b : Entrées au clavier par Questions-réponses

Modifier le programme pour que celui-ci demande les informations à l'utilisateur, puis lise les valeurs saisies au clavier

II.3 Lecture du fichier.

Ecrire dans le programme principal les codes permettant de **lire dans le fichier et afficher à l'écran** les valeurs de **nb_lentilles**, puis pour chaque lentille **son abscisse** et sa **distance focale**.

II. 4 Calcul des images successives

Ajouter le code permettant de calculer et afficher **pour chaque image** successive :

- son abscisse,
- son ordonnée,
- la tangente de l'angle de sortie de la lentille.

L'affichage devra respecter le format donné dans le programme fonctions.c. La distance image n'est pas demandée.

II. 5 Affichage des caractéristiques du système

On veut caractériser le système optique. Selon le type de système, on définit un paramètre différent caractérisant ce que l'on appelle communément le « grossissement » :

- a- pour les lunettes, qui observent des objets lointains, on définit le **Grossissement optique** comme le rapport $\tan(a_{\text{sortie}}) / \tan(a_{\text{entrée}})$ où $a_{\text{entrée}}$ est égal à l'angle de sortie de la première lentille.
- b- pour les microscopes ou les loupes (vision directe dans l'oculaire), on définit la **Puissance** comme $\tan(a_{\text{sortie}}) / y_{\text{objet}}$

- c- pour les projecteurs, les appareils photos ou les microscopes avec capture photo ou vidéo (donc projection sur un « écran »capteur), on définit le **Grandissement transversal** comme $y_{\text{image}}/y_{\text{objet}}$

Compléter le programme pour qu'il affiche ces trois paramètres (quel que soit le type de système).

II.6 Tester la dernière version du programme sur tous les fichiers téléchargés.

Les valeurs de test recommandées sont fournies dans le fichier test_lentilles.sh. Ce fichier contient les commandes qui devraient être tapées dans le terminal pour tester le programme si on a choisi l'utilisation des arguments en ligne de commande à la question II.2. On peut exécuter ces commandes soit en les tapant dans le terminal, soit en les copiant/collant (si le terminal le permet), soit en lançant ce script par la commande `bash test_lentilles.sh` (bash est l'interpréteur qui exécute les commandes tapées dans le terminal).

Vérifier que la lunette astronomique retourne l'image, et que la lunette terrestre la visualise à l'endroit. Vérifier que le projecteur agrandit bien les images.

III DESSIN DES SYSTEMES OPTIQUES

Le programme précédent permet d'évaluer les performances des systèmes optiques sous forme de texte affiché à l'écran. On aimerait avoir une représentation du système optique pour mieux comprendre son fonctionnement. On va donc modifier le programme précédent pour qu'il génère, en plus du texte, une image représentant le système optique.

Pour réaliser les images, on utilise généralement une bibliothèque graphique (il en existe énormément) qui permet soit d'afficher à l'écran directement, soit d'enregistrer une image dans un fichier. Malheureusement, l'apprentissage d'une telle bibliothèque est souvent un peu long du fait de leur polyvalence. Plutôt que d'apprendre une bibliothèque complexe, nous allons utiliser 3 fonctions déjà écrites (portables sur n'importe quelle machine) qui vont permettre de générer un fichier au format SVG (Scalable Vector Graphics). Ce format, de type XML est standardisé, portable, de type vectoriel (et pas bitmap) en format texte lisible humainement. Il est reconnu par tous les navigateurs internet modernes.

III.1 Fonction graphiques SVG

- a- Télécharger (si ce n'est pas encore fait), compiler et exécuter le programme `demo_svg.c`. Pour visualiser le fichier de sortie `demo.svg`, vous pouvez l'ouvrir avec Firefox. S'il est modifié, un fichier ouvert sous Firefox peut être rafraîchi manuellement dans le navigateur

Vous pouvez aussi regarder son contenu avec un éditeur de texte. Dans ce fichier, chaque segment est codé dans un élément XML de la forme `<line x= " " .... />`

- b- Modifier le programme `demo_svg.c` pour qu'il trace un carré vert englobant la figure géométrique précédente.

III.2 Préparation du programme de dessin des lentilles

- a- Enregistrer le programme **lentilles.c** sous **figure_lentilles.c** et modifier l'en-tête en conséquence.

- b- Copier les 3 fonctions du programme `demo_svg.c` dans `figure_lentilles.c`
- Modifier la fonction principale pour que :
- c- le programme prenne en plus comme arguments (ou demande à l'utilisateur) le nom du fichier de sortie au format SVG, et la distance maximale visible sur l'abscisse
 - d- il ouvre et ferme le fichier SVG, écrive l'en-tête (header) et la fin (footer) du fichier (dans le bon ordre bien sûr). La zone de tracé du dessin sera un carré centré sur (0,0) dont les côtés seront à la « distance maximale visible sur l'abscisse » du centre.
 - e- Faire tracer l'axe des abscisses (épaisseur 1mm en gris).

A partir d'ici, pour faire les tests, il est recommandé d'utiliser les valeurs fournies dans le fichier `test_figure_lentilles.sh` (qui peut être utilisé comme le précédent, et contient les mêmes exemples)

III.3 Dessin des lentilles

- a- Ecrire une fonction `dessiner_lentille` qui prend comme argument le fichier de tracé, l'abscisse d'une lentille et sa distance focale, et qui trace une lentille. Le tracé de la lentille ressemblera vaguement à celui des figures 1 et 2. Il sera par exemple constitué de 5 traits. Recommandations : « diamètre » lentille = 5 cm, épaisseur trait = 1mm, couleur : grise. Utiliser cette fonction dans la fonction principale pour tracer toutes les lentilles.
- b- Exécuter le programme sur tous les exemples (utilisez `bash test_figure_lentilles.sh`) et vérifier les dessins réalisés.

III.4 Dessin de l'objet, des images et des rayons

Dans la fonction principale, sur le modèle de la figure 2 :

- a- Faire tracer l'objet sous forme d'un trait rouge épais (5 mm),
- b- Faire tracer toutes les images sous forme d'un trait jaune épais (5 mm),
- c- Ecrire une fonction `tracer_rayons_construction()` qui prend le fichier de sortie, les coordonnées d'un « objet » intermédiaire et de son image, ainsi que l'abscisse de la lentille, et qui trace les trois rayons de construction comme sur la figure 1 (les rayons s'arrêteront à l'image, il n'auront pas de flèche), en blanc (1mm).
- d- Avec cette fonction, faire tracer tous les rayons lumineux indiqués sur la figure 2.
- e- Exécutez le programme sur tous les exemples.

III.5 Optimisation du code (2 points bonus)

- a- Modifier la fonction `dessiner_lentille` pour qu'elle représente différemment les lentilles convergentes et les lentilles divergentes.
- b- Lorsqu'une image est virtuelle, le tracé des rayons n'est pas très lisible. Modifier le code de `tracer_rayons_construction()` pour qu'il détecte les images virtuelles (par sa position par rapport à la lentille) et qu'il trace le prolongement des rayons de l'autre côté de la lentille.