

Travaux Pratiques - Langage C

1 Saisie du programme

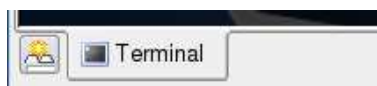
Faites la saisie du programme en vous aidant du squelette vu en TD et en y incorporant les fonctions que vous avez écrites.

Ouvrez une **konsole** et compilez le programme. L'exécutable s'appellera `modes`.

- Quelle ligne de commande utilisez-vous pour compiler le programme ?
- Exécutez le programme.
- Quels fichiers ont été créés ? Quelles sont leurs tailles ? Quelle est la signification physique de leur contenu ?

2 Utilisation de `gnuplot`

Commencez tout d'abord par créer une nouvelle session de konsole en cliquant à gauche de l'onglet de votre console :



Lancez ensuite le logiciel GnuPlot en saisissant `gnuplot` dans la console que vous venez de créer. GnuPlot est maintenant prêt à recevoir vos instructions. Commençons par représenter le contenu de `module1` en 3D. Saisissez :

```
splot "module1.txt"
```

Vous obtenez alors une représentation graphique 3D de la matrice `module1` où la surface est représentée par un nuage de points. Pour changer l'angle de vue, cliquez sur le graphique, et sans relâcher le bouton de la souris, déplacez-vous. Lorsque l'angle de vue vous convient, relâchez le bouton de la souris. Pour zoomer le graphique, cliquez avec le bouton central de la souris¹, puis déplacez la souris vers la gauche pour vous éloigner du graphique, et vers la droite pour vous en rapprocher. Toujours en cliquant avec le bouton central de la souris, déplacez-vous vers le haut ou vers le bas. Vous modifiez alors le zoom suivant l'axe vertical uniquement.

Le nuage de points n'est pas forcément le mode de représentation le plus pratique pour une surface. Nous allons relier chaque point du graphique avec ses voisins par des lignes. Saisissez la ligne suivante dans la console GnuPlot :

```
splot "module1.txt" with lines
```

Le graphique est mis à jour, et la surface est représentée par des « fils de fer » (la surface est transparente). Ce mode de représentation peut parfois être gênant pour bien apprécier tout le volume de la surface représentée. Il est généralement plus pratique d'avoir une surface pleine. Vous pouvez obtenir ce mode de représentation avec les instructions suivantes :

1. Si votre souris ne dispose pas de bouton central, cliquez simultanément sur les boutons droit et gauche de votre souris

```
set hidden3d
splot "module1.txt" with lines
```

La surface est maintenant non transparente. Pour la rendre à nouveau transparente, utilisez les commandes suivantes :

```
unset hidden3d
splot "module1.txt" with lines
```

- Tracez successivement `module1.txt` et `module2.txt`. Expliquez l'évolution du profil de l'onde laser au fil des itérations. Quels sont les largeurs des modes dans les deux plans ? Sont-ils conjugués au sens de Fourier ?
- Tracez maintenant `phase1.txt` et `phase2.txt` et commentez les graphiques. Que constatez-vous ? Donnez une explication.

3 Suppression des sauts de phase

Modifiez le contenu de la fonction `exporte_colonne_phase` de la façon suivante :

```
phase = gsl_complex_arg(gsl_vector_complex_get( vecteur , j));
if (phase<0) {phase+=M_PI;}
else if (phase>2) {phase-=M_PI;}
gsl_matrix_set( matrice , j , i , phase);
```

- Expliquez le rôle des deux lignes qui ont été ajoutées.
- Recompilez le programme et exécutez-le à nouveau.
- À l'aide de `gnuplot`, observez `phase1.txt` et `phase2.txt`. Que constatez-vous ?
- Quelle est la signification physique d'une phase plane ? Est-ce cohérent avec la condition de résonance dans la cavité laser ?
- Relancez le programme plusieurs fois et observez `phase1.txt` et `phase2.txt`. Que constatez-vous ? L'algorithme de suppression des sauts de phase est-il optimal ? Proposez une amélioration (mais ne la codez pas dans votre programme).

4 Petite remarque à propos du bruit

L'amplitude de l'onde augmentant rapidement au fil des itérations dans la cavité laser, il est difficile de voir exactement ce qu'il se passe pendant les premières itérations. Nous allons modifier légèrement le programme pour observer ceci.

- Dans un premier temps, mettez en commentaire les lignes faisant appel aux fonctions `gain` et `pertes` dans la boucle principale.
- Puis, après chaque transformation de Fourier, normalisez le vecteur `champ` à l'aide de la fonction `normalisation`.
- Recompilez le programme et exécutez-le.
- À l'aide de `gnuplot`, observez les profils de `module1.txt` et `module2.txt`. Qu'observez-vous sur le profil de `module2.txt` ?
- Pourquoi n'observez-vous pas la même chose sur `module1.txt` ?
- Modifiez la fonction `initialise_bruit` pour faire disparaître ce phénomène.
- Recompilez et exécutez votre programme. Vérifiez que ce phénomène a effectivement disparu.
- Supprimez les commentaires devant les fonctions `gain` et `pertes`, et commentez les lignes normalisant le vecteur `champ`.

5 Étude de la taille des profils : largeur à mi-hauteur

L'objectif de cette partie est d'étudier comment évoluent les tailles des faisceaux lasers dans les deux plans en fonction de la taille des filtres. Les faisceaux étant gaussiens, nous caractériserons leur dimensions grâce à leur largeur à mi-hauteur, notée FWHM².

Recopiez à la fin du programme la fonction suivante, calculant la largeur à mi-hauteur du champ pour la dernière itération.

```
/* *****
 *                               *
 *          Largeur à mi hauteur          *
 * ***** */

double FWHM( gsl_matrix* matrice )
{
    int i;
    int point1=0, point2=0;
    double max;
    gsl_vector *champ = gsl_vector_calloc( N );

    /* on place la dernière itération dans le vecteur champ */

    for(i=0;i<N;i++)
        gsl_vector_set( champ , i , gsl_matrix_get( matrice , i , ITERATIONS-1 ));

    /* puis on recherche le maximum de champ */

    max = gsl_vector_max( champ );

    /* recherche du premier point à mi-hauteur */

    while(gsl_vector_get( champ , point1) < max/2.0)
        point1++;

    /* recherche du second point à mi-hauteur */

    point2 = point1+1;

    while(gsl_vector_get( champ , point2) > max/2.0)
        point2++;

    return (point2-point1);
}
```

- Expliquez le fonctionnement de cette fonction
- Pourquoi travaille-t-on sur la colonne `ITERATIONS-1` de la matrice ?
- Comment la largeur à mi-hauteur est-elle déterminée ?
- Dans quelle unité est-elle donnée ?

À la fin de la boucle principale, appelez la fonction FWHM pour calculer la largeur à mi-hauteur du faisceau au niveau du miroir 1 et au niveau du miroir 2.

Remplissez alors le tableau suivant :

2. pour Full Width at Half Height

ΔX_1 (mm)	ΔX_2 (mm)	FWHM ₁ (mm)	FWHM ₂ (mm)	FWHM ₁ \times FWHM ₂ (mm ²)
0.5				
1				
1.5				
2				
3				
4				
5				
10				
15				
20				

- Que constatez-vous ? Est-ce cohérent ?
- Quel est l'avantage d'avoir utilisé un échantillonnage variable dans les plans des miroirs ?

6 Filtrage un peu plus élaboré

Modifiez les filtres comme le montre le schéma ci-dessous. Compilez et exécutez plusieurs fois votre programme. Commentez.

