Université Paris Diderot – Paris 7
UFR de Physique
L3 MAGISTERE DE PHYSIQUE

Projet de physique expérimentale

Diffraction optique de structures quasi-périodiques

Projet de physique expérimentale du

Magistère de Physique

Année 2007-2008

Par

Olivier GAZZANO

Romain RONCIN

Juliette VOYEZ

Diffraction optique de structures quasi-périodiques

Table des matières

Introduction **1**

1 Structures périodiques, quasi-périodiques et apériodiques **2**
1.1 Approche mathématique . . . 2
1.2 Comment générer une structure quasi-périodique ? . . . 3
1.3 Symétries autorisées . . . 4
1.4 Structures quasi-périodiques et symétries interdites . . . 5

2 Création des grilles au format PostScript **6**
2.1 Pourquoi utiliser le PostScript ? . . . 6
2.2 Comment générer un fichier PostScript à partir d'un programme en C ? . . . 6
2.3 Utilisation du PostScript . . . 6
2.4 Réalisation d'une grille de points . . . 7
2.5 Réalisation d'une grille de lignes horizontales ou verticales . . . 7
2.6 Grilles ayant un angle quelconque . . . 8
2.6.1 Présentation du problème . . . 8
2.6.2 Solution possible . . . 8
2.6.3 Algorithme . . . 9
2.6.4 Création de la page . . . 10

3 Expérience **11**
3.1 Qu'est-ce que la diffraction ? . . . 11
3.2 Le montage en lui-même . . . 11
3.3 Les grilles . . . 13
3.4 Les mesures . . . 13

4 Etude théorique de la diffraction et analyse des résultats **17**
4.1 Fentes de Young . . . 17
4.2 Diffraction par N fentes linéaires . . . 18
4.2.1 Cas général . . . 18
4.2.2 Fentes équidistantes . . . 19
4.2.3 Pertubation sinusoïdale, approximation des trois tâches . . . 21
4.3 Diffraction par N trous dans un plan . . . 23
4.3.1 Cas général . . . 23
4.3.2 Trous régulièrement répartis . . . 24
4.4 Diffraction par des surfaces quasi-périodiques . . . 28

Conclusion **30**

A Bibliographie **31**

B Logiciels utilisés **31**

C Diffraction par un programme en C **32**

D Générer une suite de L et de S suivant une suite de Fibonacci **33**

E Programme en C permettant de générer une figure comportant 5 motifs croisées ayant un angle de 72° entre elles **34**

F Page comportant les grilles générées **38**

Table des figures

1 Construction d'une séquence unidimensionnelle quasi-périodique . . . 2
2 Reproduction des noeuds d'un réseau . . . 4
3 Pavage de Penrose . . . 5
4 Exemple de commandes PostScript . . . 6
5 Grille de points . . . 7
6 Grille de lignes . . . 7
7 Exemple de dépassement du cadre de l'image . . . 8
8 Intersections possibles entre une droite et un cadre . . . 8
9 Exemple de dessin comportant 5 motifs de lignes ayant un angle de 72° . . . 10
10 Positionnement d'une grille sur la page . . . 10
11 Schéma du montage . . . 11
12 Photo du montage . . . 11
13 Photos du montage avec le vidéoprojecteur . . . 12
14 Courbe d'étalonnage (inverse de la largeur de fente) . . . 13
15 Grilles de diffraction . . . 14
16 Photo d'une figure de diffraction . . . 14
17 Courbes d'étalonnage . . . 15
18 Diffraction par les fentes de Young . . . 17
19 Diffraction par N fentes . . . 18
20 Diffraction par N fentes équidistantes . . . 19
21 Simulation de la diffraction par 30 fentes équidistantes . . . 20
22 Photo de la figure de diffraction d'une grille de lignes verticales et régulières . . . 20
23 Image de lignes ayant une perturbation sinusoïdale et sa figure de diffraction . . . 22
24 Points sur un plan . . . 23
25 Représentation des vecteurs k_i et k_r dans l'espace . . . 23
26 Point sur une grille plane régulière . . . 24
27 Représentation du vecteur $\mathbf{k_i}$ . . . 25
28 Simulation de la diffraction par des points sur une grille plane . . . 25
29 Détail de la figure 28 . . . 26
30 Courbe 3D issue de la simulation de la diffraction par des points sur une grille plane . . . 26
31 Grille de diffraction 0°-90° . . . 27
32 Photo de diffraction de la figure 31 . . . 27
33 Grille et photo de diffraction obtenue grâce à une grille de Fibonacci . . . 28
34 Grille et figure de diffraction obtenue grâce à une grille de Fibonacci 0-90° . . . 28
35 Grille et figure de diffraction de grilles ayant un angle de 72° entre elles avec la séquence de Fibonacci . . . 29
36 Page comportant les grilles générées . . . 38

Introduction

Dans notre précédent travail, nous avions exposé les caractéristiques générales des quasi-cristaux, et notamment travaillé sur les symétries interdites que présentent ces derniers. Nous avions aussi travaillé sur la théorie de la diffraction, à deux puis N diffuseurs. Cependant, cette étude théorique n'était pas applicable à des structures quasi-périodiques. Dans la seconde partie de notre projet, nous allons réaliser une approche expérimentale de la diffraction de la lumière par des surfaces quasi-périodiques.
Dans une première partie, nous expliquerons comment générer une structure quasi-périodique et l'intérêt de la séquence de Fibonacci. Nous rappellerons aussi le principe des symétries interdites et autorisées. Puis, nous parlerons de la réalisation des grilles de diffraction, à une et deux dimensions, périodiques et quasi-périodiques, à l'aide de programmes en langage C . Par la suite, nous décrirons nos expériences et les mesures effectuées. Enfin, nous confronterons la théorie de la diffraction avec l'analyse de nos résultats.

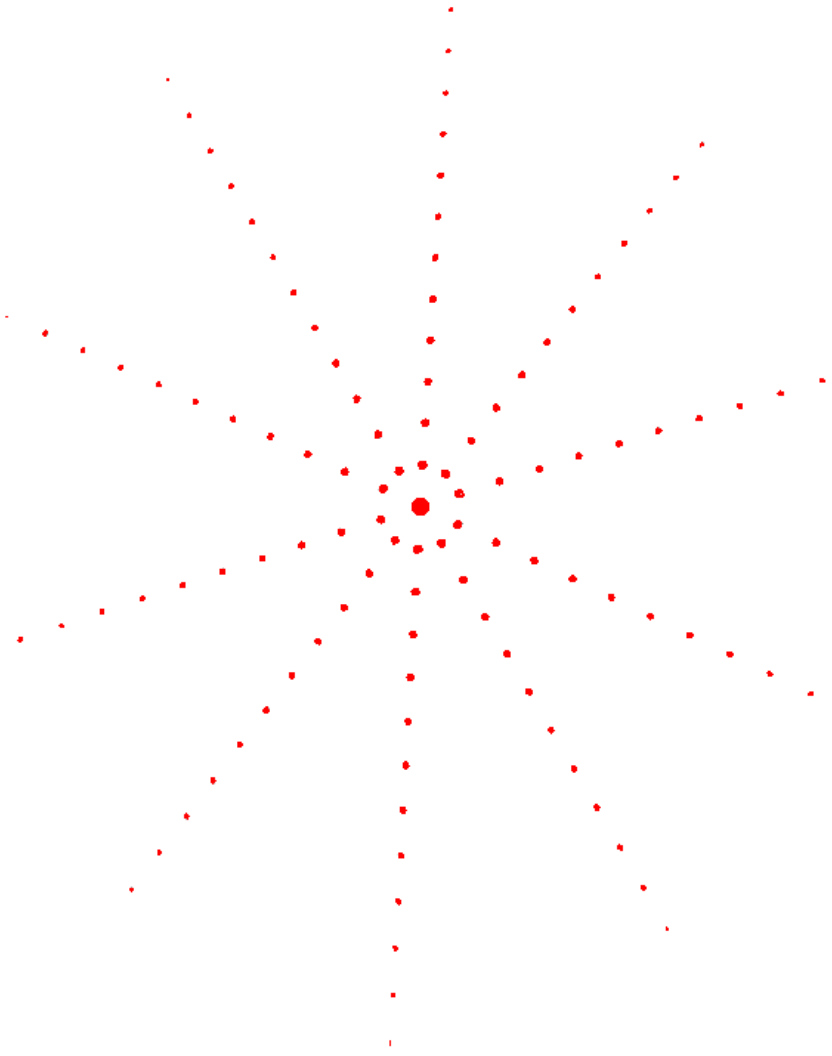


Photo d'une figure de diffraction de 5 grilles régulières ayant un angle de 72° entre elles.
Le noir du fond a été transformé en blanc, et le contraste a été amélioré afin de faire disparaitre le bruit pour mieux voir les taches.

1 Structures périodiques, quasi-périodiques et apériodiques

1.1 Approche mathématique : intersection d'un espace de dimension $n \leq$ avec un plan de dimension 2

L'objectif de ce paragraphe est de comprendre la différence entre une structure périodique, quasi-périodique et apériodique. Pour cela, on va donner une explication mathématique.

On dessine dans le plan un réseau carré simple que l'on intersecte par une droite quelconque de pente $\tan(\alpha)$ irrationnelle par rapport aux directions de base x et y du réseau (figure 1). On collecte tous les carrés élémentaires qui sont intersectés par cette droite ; on obtient ainsi une bande de carreaux en escalier où les marches se répartissent de façon uniforme mais non périodique, la pente de la droite étant irrationnelle. On choisit pour chaque carré intersecté un point représentatif, par exemple le sommet supérieur droit, qu'on projette orthogonalement sur la droite. C'est la «méthode des cages » proposé par John Conway.

On obtient ainsi une séquence de segments longs et courts correspondants, selon l'orientation choisie, aux projections sur la droite des sommets bordant les arêtes selon x et selon y des carrés. Un objet périodique du plan a engendré, par coupe d'orientation **irrationnelle** d'une droite, un objet **quasi-périodique** à une dimension.

La quasi-périodicité repose sur l'irrationalité de la pente de la droite. Dès que la pente est **rationnelle**, la séquence est **périodique**. Un objet quasi-périodique correspond ainsi à une interpolation irrationnelle entre objets périodiques. A une dimension, on peut en engendrer une infinité non dénombrable ($\mathbb{Q}$ est indénombrable) alors que les séquences périodiques sont, elles, dénombrables ($\mathbb{N}$ et $\mathbb{Z}$ sont indénombrables).

Toute séquence quasi-périodique peut être vue comme la limite d'une suite de séquences périodiques de période croissante construites à partir d'un faisceau de droites dont les pentes (rationnelles) convergent vers le nombre irrationnel caractéristique de la séquence quasi-périodique limite. La proportion de segments courts par rapport aux longs dépend de la pente de la droite et la distribution relative entre longs et courts correspond, par construction, au mélange optimal. Les quasi-cristaux sont aux cristaux ce que les nombres irrationnels sont aux nombres rationnels : ils viennent enrichir la panoplie des solides ordonnés à longue distance et partagent avec les cristaux la totalité de leurs propriétés géométriques, sauf la périodicité.

Aujourd'hui l'algorithme générateur de quasi-cristaux consiste, à partir d'un réseau dans un espace de dimension N>d (ici N=2, d=1), à placer orthogonalement à un hyperplan de dimension d, irrationnellement orienté par rapport au réseau, une collection de volumes bornés de dimension N-d et d'en collecter les intersections avec l'hyperplan : on obtient un ensemble de points qui constituent un pavage quasi-périodique de l'hyperplan, comme par exemple un pavage de Penrose (figure 3).

Pour ce qui est des structures apériodiques, il n'existe ni espace ni surface engendrant cette structure.

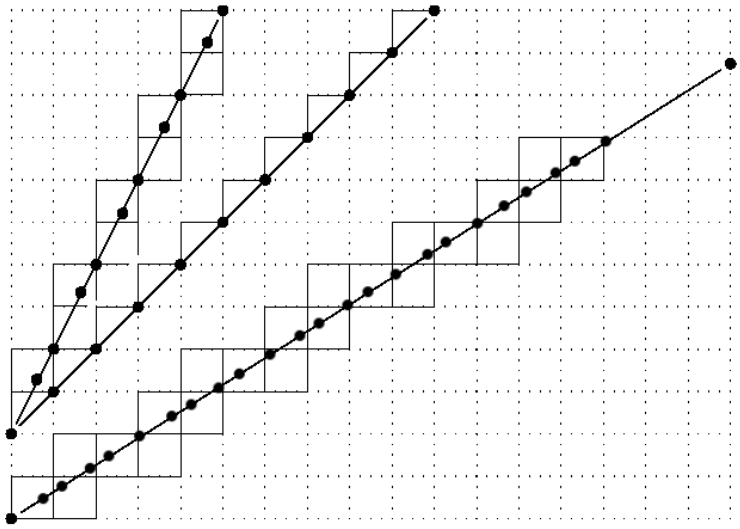


FIG. 1 – **Principe de construction d'une séquence unidimensionnelle quasi-périodique. (1) on trace une droite de pente irrationnelle sur un réseau carré ; (2) on collecte les sommets supérieurs droits des carrés intersectés qu'on projette orthogonalement sur la droite ; (3) la séquence de points ainsi obtenue est quasi-périodique et peut être engendrée directement par intersection de petits segments verticaux copiés en chacun des nœuds du réseau avec la droite : c'est la méthode dite de « coupe ».**
Les deux droites du haut ont un angle rationnel, alors que la troisième à un angle irrationnel.

1.2 Comment générer une structure quasi-périodique plane ? Utilisation de la séquence de Fibonacci

La «méthode des cages » expliquée précédement permet de genérer toutes les structures quasi-périodiques. Par contre cette méthode est difficile à mettre en place informatiquement. De plus, pour les structures unidimentionnelles, il existe une autre méthode qui utilise la séquence de Fibonacci.

En 1202, Fibonacci introduit cette suite qui aujourd'hui porte son nom. Celle-ci est définie par la relation de récurrence suivante :

$$F_{n+1} = F_n + F_{n-1}$$

avec les conditions initiales : $F_0 = 0$ et $F_1 = 1$

Cette suite est d'autant plus célèbre qu'elle permet de retrouver le nombre d'or. En effet, la limite du quotient F_{n+1}/F_n se rapproche du nombre d'or lorsque n tend vers l'infini : $\lim_{n\to+\infty} \frac{F_{n+1}}{F_n} = \frac{1+\sqrt{5}}{2}$

Pour obtenir les différents termes de la suite de Fibonacci, on utilise la matrice P définie par $P = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$ et on introduit une population initiale : $\vec{p_0} = (F_1, F_0) = (1, 0)$

Pour la première génération, on a alors :

$$(1,0)\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} = (1,1)$$

Pour la deuxième génération, on a :

$$(1,1)\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} = (1,0)\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^2 = (1,0)\begin{pmatrix} 2 & 1 \\ 1 & 1 \end{pmatrix} = (2,1)$$

Pour la troisième génération, on a :

$$(2,1)\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} = (1,0)\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^3 = (1,0)\begin{pmatrix} 3 & 2 \\ 2 & 1 \end{pmatrix} = (3,2)$$

Ainsi, on montre que pour la (n+1) génération, on a :

$$\vec{p_0}P^{n+1} = (1,0)\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^{n+1} = (1,0)\begin{pmatrix} F_{n+1} & F_n \\ F_n & F_{n-1} \end{pmatrix} = (F_{n+1}, F_n)$$

On utilise ensuite la suite de Fibonacci pour générer des grilles quasi-périodiques. En effet, la séquence de Fibonacci peut se construire directement si l'on pose la substitution suivante :

$$L \longrightarrow LS$$
$$S \longrightarrow L$$

en prenant L pour un segment long et S pour un segment court.

On passe ainsi d'une génération à l'autre en substituant LS pour chaque L et L pour chaque S. Si on commence avec un segment court, voilà ce qui se passe après plusieurs substitutions :

$$S$$
$$L$$
$$LS$$
$$LSL$$
$$LSLLS$$
$$LSLLSLSL$$
$$LSLLSLSLLSLLS$$
$$LSLLSLSLLSLLSLSLLSLSL$$

On retrouve bien une séquence similaire à celle que l'on peut retrouver avec la méthode précédente. En effet, si on reprend la troisième droite de la figure 1, on trouve la séquence : LSLSLLSLSLLSLSLLSLLSLSL.

En annexe D, se trouve le programme permettant de générer cette séquence.

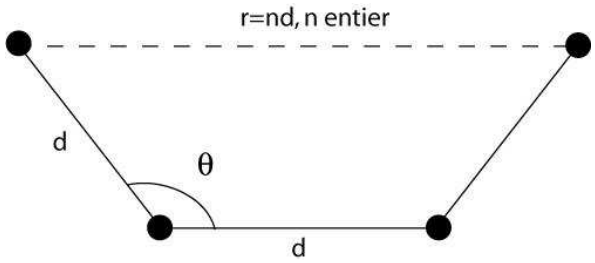


FIG. 2 – **Reproduction des noeuds d'un réseau**

1.3 Symétries autorisées

On rappelle ici quelles sont les symétries autorisées dans un pavage plan périodique.
Soit d la plus petite distance entre deux noeuds du réseau. Une rotation d'angle θ du réseau conduit à deux nouveaux noeuds qui doivent se superposer à des noeuds du réseau pour que la rotation soit une opération de symétrie. Une rapide analyse géométrique conduit alors à la conclusion que le réseau reste invariant uniquement pour des valeurs particulières de θ. On va démontrer ainsi que seules les symétries d'ordre 2, 3, 4 et 6 sont permises :
Comme nous sommes dans une structure périodique :

$$\exists n \in \mathbb{Z} \text{ tel que } r = nd$$

Ainsi,

$$2d\sin(\theta - \frac{\pi}{2}) + d = nd = r$$

$$\sin(\theta - \frac{\pi}{2}) = \frac{n-1}{2}$$

$$\cos(\theta) = \frac{n-1}{2}$$

$$-1 \leq cos(\theta) = \frac{n-1}{2} \leq 1$$

On peut alors dresser le tableau suivant, avec en dernière ligne les différents ordres de symétries permis pour obtenir un pavage périodique :

n	-1	0	1	2	3
$\cos(\theta)$	-1	$-\frac{1}{2}$	0	$\frac{1}{2}$	1
θ	$\frac{2\pi}{2}$	$\frac{2\pi}{3}$	$\frac{2\pi}{4}$	$\frac{2\pi}{6}$	0
Ordre de la symétrie	2	3	4	6	1

Un pavage périodique avec une symétrie d'ordre 5 ne peut par conséquent pas être envisagé.

1.4 Structures quasi-périodiques et symétries interdites

Un bon exemple de pavage quasi-périodique est celui en deux dimensions réalisé par Roger Penrose (figure 3). Ces structures quasi-périodiques ont la propriété de présenter un spectre de diffraction essentiellement discret, tout comme celui des structures périodiques. Cependant, dans les pavages tout comme dans les figures de diffraction, on remarque alors la présence de symétries interdites dans les structures périodiques "classiques", d'ordre 5 ou 10, par exemple. C'est notamment le cas dans les quasi-cristaux, comme le AlPdMn, ou le AlNiCo.

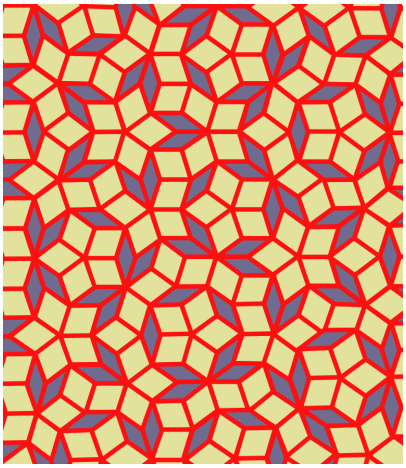


FIG. 3 – **Pavage de Penrose**

2 Création des grilles au format PostScript

2.1 Pourquoi utiliser le PostScript ?

Nous voulons créer des grilles pouvant être imprimées à une très haute résolution d'impression (12 000 dpi). Pour cela il faut une image qui peut être grossie à l'infini sans perdre de qualité.

D'autre part, nous voulons aussi réaliser des simulations grâce au logiciel ImageJ qui fait la transformée de Fourier des images. Il nous faut, ici encore, des images de grandes dimensions et donc de grande qualité si l'on veut un résultat convenable.

Le dessin vectoriel est une solution. Nous avons choisi le PostScript® (.ps).

Pour cela, on utilise un programme écrit en C qui crée un fichier de sortie directement au format PostScript.

L'objectif de cette partie va être de détailler comment générer des figures de diffraction en PostScript à partir d'un programme en C.

2.2 Comment générer un fichier PostScript à partir d'un programme en C ?

Une solution permettant de générer un fichier PostScript (ps) à partir d'un programme en C est d'utiliser la fonction «ofstream» sur un fichier que l'on peut appeler par exemple «fentes.ps» lors de l'appel de la fonction sortie.

A l'entête du programme en C, il y aura de ce fait :

$$ofstream\ sortie("fentes.ps");$$

Lors de l'exécution de programme, la commande

$$sortie << commande\ ps << endl;$$

permettra d'écrire «commande pour le ps» dans le fichier ps et d'aller à la ligne (avec «endl»).

2.3 Utilisation du PostScript

Le PostScript est un langage de programmation qui permet notamment de faire du dessin. Il y a donc des commandes spécifiques à ce langage. Voila quelques commandes que l'on utilisera pour afficher nos figures de diffraction :

- « newpath » : initialise la commande de dessin
- « x y moveto » : déplace le curseur à l'emplacement donné par deux nombres x et y. L'abscisse du point est x et y est ordonnée.
- « x' y' lineto » : trace un trait depuis la position du curseur jusqu'au point (x',y').
- « stroke » : termine la zone de dessin.
- « x y r 0 θ arc » : affiche une arc de cercle de centre (x,y) de rayon r et d'angle θ.

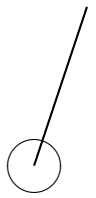


FIG. 4 – Exemple de commandes PostScript :

newpath
0 1 *moveto*
1 4 *lineto*
0 1 2 0 360 *arc*
stroke

2.4 Réalisation d'une grille de points

Pour générer une grille de points régulièrement répartis sur un plan, il faut utiliser des boucles.

```
for ( int i=1 ;i<80 ;i++ ){
for ( int j=1 ;j<80 ;j++ ){
sortie«"newpath"«endl;
sortie«i«" "«j«" 0.5 "«" 0 "«" 360 "«" arc"«endl;
sortie«"fill"«endl;
sortie«"stroke"«endl«endl;
cmp++;
} }
```

Cette boucle va afficher des cercles pleins (grâce à la commande «fill») à tous les points (i, j) où $0 \leq i < 80$ et $0 \leq j < 80$. Comme les cercles ont un diamètre très faible, ils peuvent être considérés comme ponctuels.
Après compilation du programme en C et en affichant le ficher .ps grâce à un logiciel comme kghostview ★[5], on obtient :

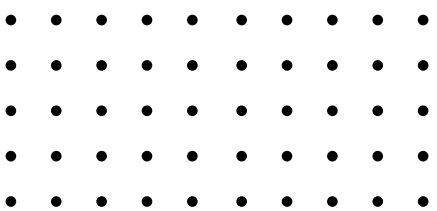


FIG. 5 – **Extrait d'une grille de points ronds.**

2.5 Réalisation d'une grille de lignes horizontales ou verticales

Pour construire une grille de diffraction comportant des lignes horizontales ou verticales, il faut aussi utiliser des boucles. Comme on l'a déjà vu, il suffit d'avoir les coordonnées des points de départ et des points d'arrivée pour pouvoir tracer une ligne.

Pour tracer des lignes horizontales (respectivement verticales), il faut faire varier les y (respectivement les x) et fixer l'autre variable.

```
for (int j=1 ;j<80/0.1 ;j++){
sortie«"n"«endl;
sortie«a*j«" "«0«" m"«endl;
sortie«a*j«" "«80«" l"«endl;
sortie«"s"«endl;
cmp++;
} }
```

Dans cet exemple, on trace des lignes verticales espacées de la distance a et ayant une longueur de 80.

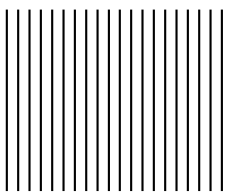


FIG. 6 – **Extrait d'une grille de lignes verticales.**

2.6 Grilles ayant un angle quelconque

2.6.1 Présentation du problème

Nous allons maintenant construire une grille ayant des fentes parallèles mais formant un angle θ avec l'horizontal. Nous voulons que l'image obtenue soit dans une cadre et qu'elle ne dépasse pas.

Le problème ici vient du fait qu'il n'est pas possible de faire tourner le repère. Relier des points des deux axes nous fait sortir du cadre de l'image. La figure 7 illustre ce problème. De plus, selon l'angle d'inclinaison des droites, ce phénomène est plus amplifié. Il faut donc trouver une solution.

Il n'est pas possible de rajouter des lignes blanches pour masquer les lignes en trop car l'imprimante lit en linéaire le programme et donc elle imprime la ligne noire et comme elle ne peut pas imprimer le blanc alors la ligne reste sur le papier.

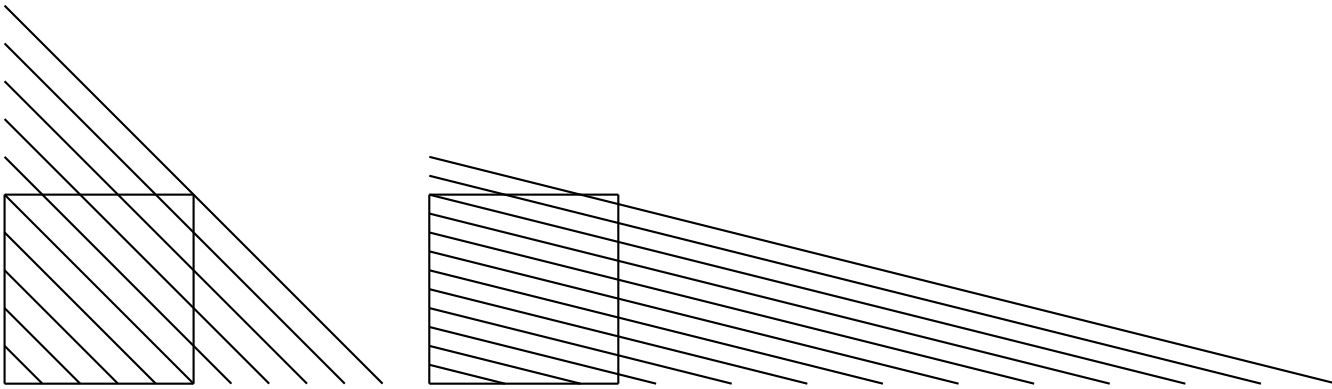


FIG. 7 – **Exemple de dépassement du cadre de l'image.**

2.6.2 Solution possible

Afin de simplifier l'étude, on considérera que la droite est strictement décroissante.
La solution consiste à calculer la coordonnée des points d'intersection entre la droite à tracer et le cadre de l'image. On distingue trois cas possibles. La figure 8 montre ces cas.

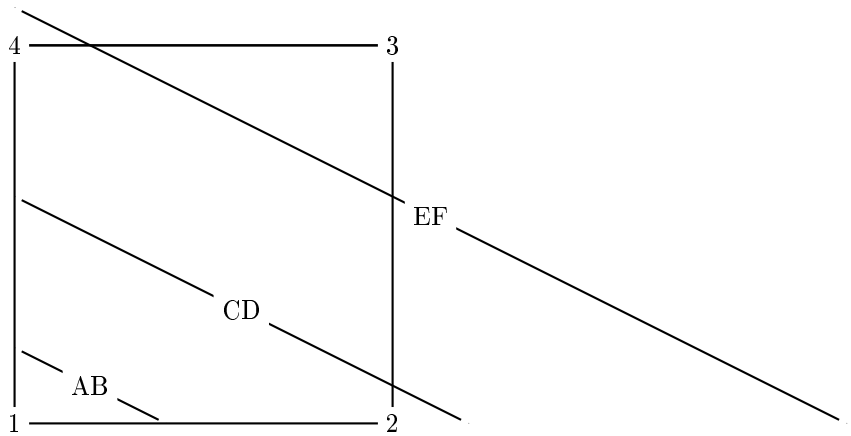


FIG. 8 – **Intersections possibles entre une droite et un cadre**

On voit que :
- Cas 1 : la droite AB coupe le côté 1-4 (on appelle ce point A) et le côté 1-2 (point B)
- Cas 2 : la droite CD coupe le côté 1-4 (point C) et le côté 2-3 (point D)
- Cas 3 : la droite EF coupe le côté 3-4 (point E) et le côté 2-3 (point F)

2.6.3 Algorithme

Considérons une droite faisant un angle θ avec l'horizontale et passant par le point $(b,0)$. Son équation dans un repère orthonormé est $y = (\tan(\theta))x + b$.
Cette droite coupe dans tous les cas le côté 1-2. Le point d'intersection (x_i, y_i) est la solution du système :

$$\left\{ \begin{array}{rcl} y_i & = & (\tan(\theta))x_i + b \\ x_i & = & 0 \end{array} \right. \quad (1)$$

Donc la solution est évidement $x_i = 0$ et $y_i = b$.
Si on est dans le cas 1 ou 2, alors y_i est forcement inférieur à la hauteur du cadre. Ce point est donc valable.
Par contre, si on est dans le cas 3, alors y_i est sorti du cadre. Ce point ne convient pas. Il faut donc prendre l'intersection avec le coté 4-1. Le point d'intersection (x_i, y_i) vérifie (h étant la hauteur du cadre) :

$$\left\{ \begin{array}{rcl} y_i & = & (\tan(\theta))x_i + b \\ y_i & = & h \end{array} \right. \quad (2)$$

Ainsi, $x_i = -\frac{b}{\tan(\theta)}$ et $y_i = h$

Remarque :
Comme par hypothèse, la droite est strictement décroissante, alors $\theta < 0$ et donc $x_i > 0$

Etudions maintenant les cordonnées de l'autre extrémité.
On calcule en premier l'intersection entre l'axe des x et la droite. Dans tous les cas, on trouve :

$$x_i = -\frac{b}{\tan(\theta)} \text{ et } y_i = 0$$

De ce fait, si $-\frac{b}{\tan(\theta)} \leq l$ avec l la largeur du cadre, alors on est forcément dans le cas 3 et donc ce point convient.
Si par contre, $-\frac{b}{\tan(\theta)} > l$ alors on est dans le cas 2 ou dans le cas 3. Il faut donc calculer la cordonnée du point d'intersection entre la droite d'étude et la droite $x = l$.
On trouve $x_i = l$ et $y_i = (\tan(\theta))\ l + b$.

Tableau récapitulatif :

Conditions	$b \leq h$	$b > h$
Si $-\frac{b}{\tan(\theta)} \leq l$	Point de départ : ($0, b$) Point d'arrivée : ($\frac{b}{\tan(\theta)}$; 0)	Point de départ : ($-\frac{b}{\tan(\theta)}$; h) Point d'arrivée : ($-\frac{b}{\tan(\theta)}$; 0)
Si $-\frac{b}{\tan(\theta)} > l$	Point de départ : ($0, b$) Point d'arrivée : (l ; $\tan(\theta)\ l + b$)	Point de départ : ($-\frac{b}{\tan(\theta)}$; h) Point d'arrivée : (l ; $\tan(\theta)\ l + b$)

A partir de ce tableau, il est possible de réaliser le programme. En annexe E, se trouve le code du programme en C permettant de générer 5 grilles croisées ayant un angle de 72° entre elles.

La figure 9 est issue de ce programme. On a utilisé les logiciels gcc[★1], kghostview[★5] et ImageMagick[★8].

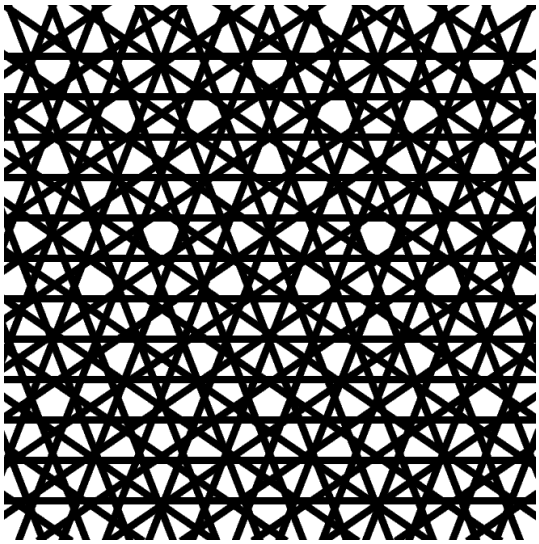


FIG. 9 – **Exemple de dessin comportant 5 motifs de lignes ayant un angle de 72°. Il a été généré avec cette méthode.**

2.6.4 Création de la page

Nous allons maintenant voir comment on peut assembler les différentes figures sur une même page.
Pour cela, il suffit de rajouter une commande de position. Introduisons pour cela deux variables, par exemple posx et posy représentant respectivement l'abscisse et l'ordonnée du coin bas gauche de la figure de diffraction. L'origine de la page est le coin en bas à gauche (figure 10). Ensuite, il faut rajouter posx ou posy à tous les endroits où l'on sort les coordonnées d'un point pour le fichier PostScript.

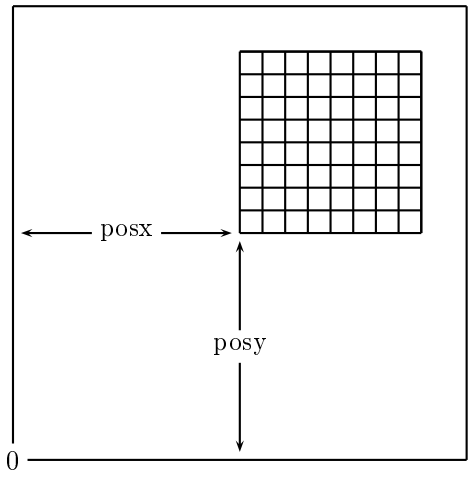


FIG. 10 – **Positionnement d'une grille sur la page**

L'annexe F comporte la page qui a été générée grâce, entre autre, aux méthodes expliquées dans cette section. Notons que le programme qui génère cette page fait plus de 3 500 lignes. Il n'était donc pas possible de joindre ce programme en annexe [1].

[1] Une copie peut être demandée et envoyée par courriel.

3 Expérience

3.1 Qu'est-ce que la diffraction ?

La diffraction est le nom donné à la déviation de la lumière. Il s'agit par conséquent de la modification du trajet d'une onde au voisinage d'un obstacle. On observe notamment qu'une onde initialement plane se propage, après la traversée d'une ouverture de petite dimension dans des directions différentes de la direction d'incidence. Cette déviation produit une figure de diffraction due à l'interférence entre les rayons de lumière parcourant différentes distances.

3.2 Le montage en lui-même

Le matériel utilisé est le suivant :
- Un laser de classe 3R 3 mW 835 nm
- Un objectif de microscope (grossissement x10)
- Une lentille de 100 mm, une de 200 mm et une de 500 mm
- Un diaphragme peut également être utilisé.

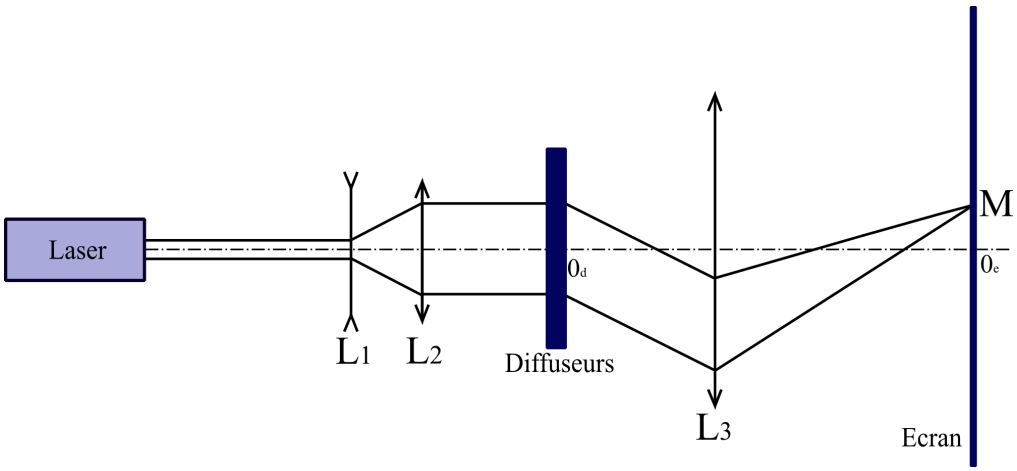


FIG. 11 – **Schéma du montage théorique**

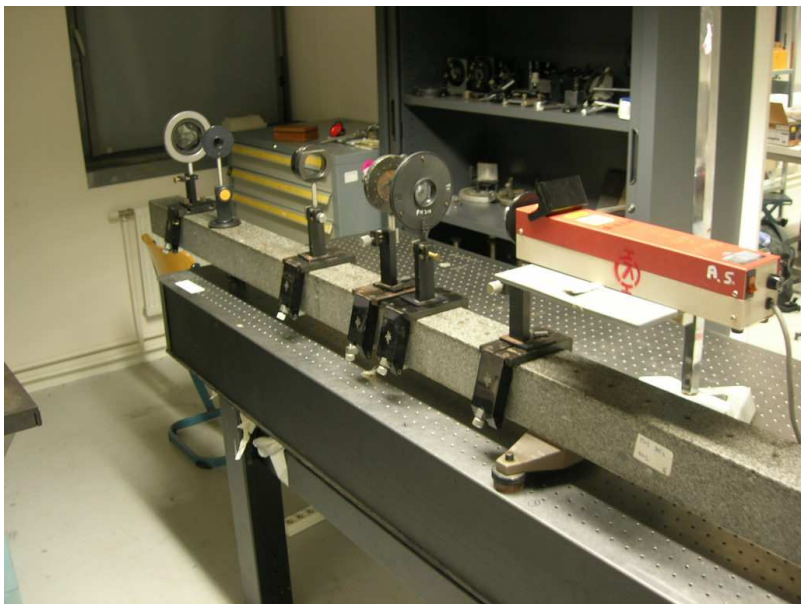


FIG. 12 – **Photo du montage**

Pour mettre en place notre montage (figures 11 et 12), nous avons tout d'abord placé un laser de classe 3R 3 mW 835 nm, puis un objectif de microscope (grossissement x10) sur le banc d'optique. Une lentille de 200 mm est par ailleurs

placée après l'objectif de microscope afin d'obtenir un faisceau de rayons parallèles à la sortie de la lentille de 200 mm. Ce faisceau peut ensuite traverser un diaphragme pour épurer son contour. L'objet à étudier, c'est-à-dire nos grilles de diffraction, sont placées à la sortie du diaphragme, la distance important peu puisqu'on a un faisceau de rayons parallèles. Enfin, une lentille de 500 mm et une lentille de 100 mm nous permettent de créer un système afocal afin d'obtenir notre image de diffraction.

Au fil des séances, nous avons tenté d'améliorer ce système :

1. Tentative d'épuration du faisceau laser par un trou de 20 μm. Cependant, l'intensité qui ressortait du montage se révèlait être trop faible pour poursuivre dans de bonnes conditions une expérience de diffraction.

2. Tentative de diffraction à l'aide d'un ordinateur relié à un vidéoprojecteur(figure 13). Le but était d'éliminer les défauts remarqués par rapport à la qualité de l'impression et du transparent. On utilise la grille créée informatiquement et on essaie de réaliser la diffraction, via le vidéoprojecteur. Cependant, la lampe du vidéoprojecteur n'étant pas une source ponctuelle de lumière, l'expérience de diffraction n'a malheureusement pas pu aboutir.

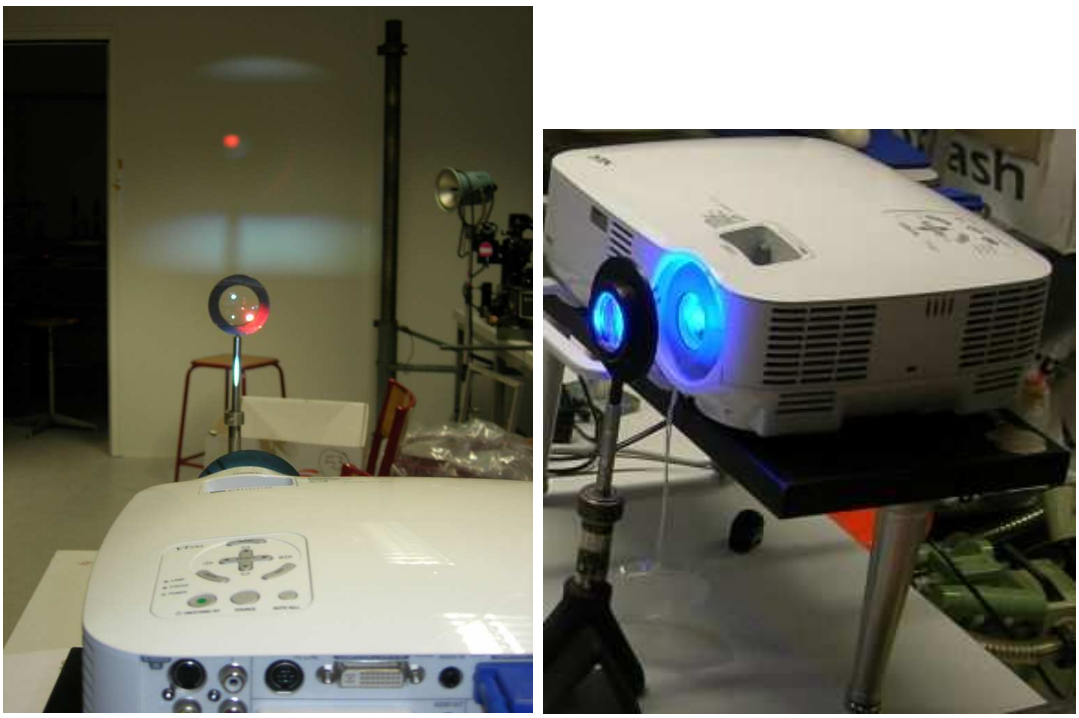


FIG. 13 – **Photos du montage avec le vidéoprojecteur**

3.3 Les grilles

Les grilles de diffraction sont notre élément principal, puisque c'est à partir de leurs figures de diffraction que les calculs et interprétations sont possibles. Ainsi, nous avons commencé par étudier des grilles simples, c'est-à-dire à des grilles périodiques en 1D, nous permettant d'étalonner notre système, avant de passer à des grilles plus complexes, dont l'interprétation est plus difficile, comme c'est le cas pour les grilles créées à partir de séquences de la suite de Fibonacci.

Les premières grilles de diffraction ont été réalisées grâce à un programme en C++ créé avec le logiciel Borland C++ 6, sous Windows. Ce programme, à partir d'un fichier image au format bitmap, réalise une grille en une dimension, composée de barres noires, toutes identiques, séparées par des espaces blancs, de largeur variable. La grille est elle même au format bitmap. Le programme commence par calculer le nième terme de la séquence de Fibonacci (où n est une valeur choisie par l'utilisateur). Puis, l'utilisateur rentre les valeurs de l'espace blanc long (L) et de l'espace blanc court (S). On obtient ainsi une grille de Fibonacci en une dimension au format bitmap. La grille est ensuite sauvegardée.

Notons qu'on peut également obtenir une grille périodique, en rentrant des valeurs de L et de S identiques dans le programme.

Pour obtenir un réseau périodique, où un réseau de Fibonacci, on fait subir une rotation de 90° à la grille, et on la superpose à la grille initiale, à l'aide d'un logiciel de dessin.

Ce programme présente l'avantage d'être facile et rapide d'utilisation. C'est pourquoi nous l'avons utilisé au début de l'expérimentation, afin de réaliser rapidement un montage optimal.

Cependant, par la suite, les grilles seront générées par un programme en C++ réalisé sous Linux, afin de sortir des grilles au format postscript.Les grilles doivent être imprimées dans des formats très petits. En effet, pour que la lumière soit diffractée par celles-ci, il faut que la largeur des fentes soit de l'ordre de la longueur d'onde du laser utilisé.

Puisque les grilles de diffraction se révèlent être au centre de nos expériences, il nous a semblé intéressant d'observer nos grilles à la loupe binoculaire, afin de se rendre compte par nous-même de leur qualité. Nous avons alors observé que sur l'imprimante qui était à notre disposition, la qualité de l'impression et celle du transparent n'étaient pas assez bonnes pour obtenir une figure de diffraction précise (figure 14) . C'est pour cela que nous avons fait appel à des professionnels pour obtenir une meilleure résolution d'impression et des transparents plus propres.

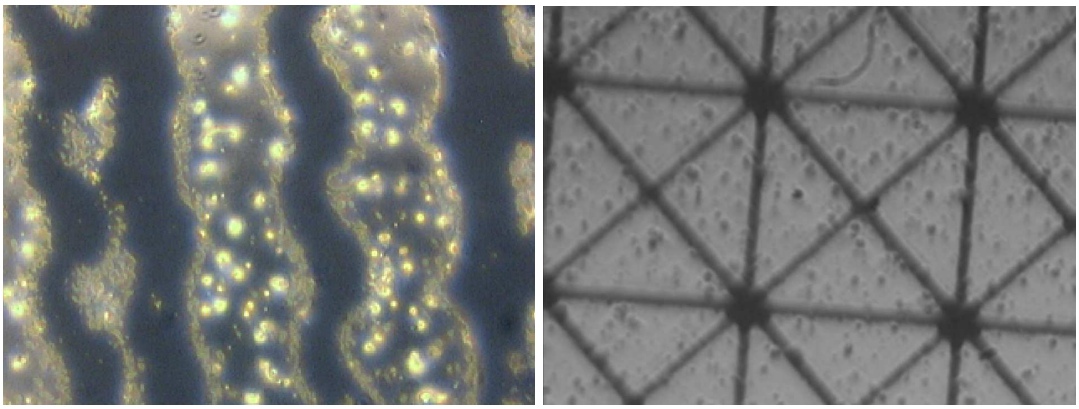


FIG. 14 – **Grilles de diffraction vue au binoculaire.**
A gauche : impression en basse résolution (environ 1200dpi)
A droite : impression en plus haute résolution (3600 dpi).

3.4 Les mesures

On étalonne notre système à partir de grilles périodiques 1D.
On utilise celles où seule la distance entre les fentes change, la largeur de la barre noire pour chacune des grilles restant la même.

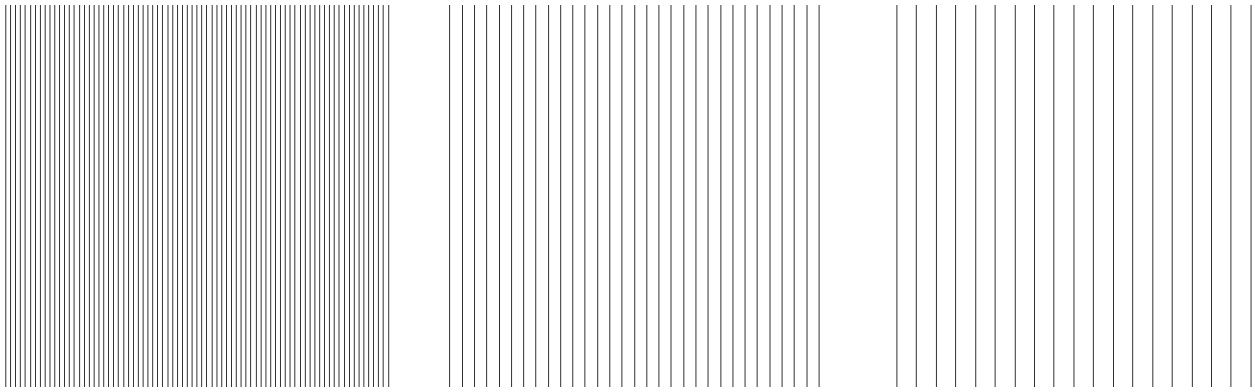


FIG. 15 – **Grilles de diffraction où seule la distance entre les fentes changent. A gauche : d = 1 pixel, au centre : d = 2,5 pixels et à droite : d = 4 pixels**

Le tableau suivant fait état des paramètres des différentes grilles utilisées pour cet étalonnage :

	Largeur barre noire (en pixel)	Distance entre les fentes (en pixel)	Largeur fente (en pixel)	Largeur fente (en m)
1+4	0,05	1	0,95	$2,14.10^{-4}$
1+5	0,05	1,5	1,45	$3,26.10^{-4}$
1+6	0,05	2	1,95	$4,39.10^{-4}$
1+7	0,05	2,5	2,45	$5,51.10^{-4}$
1+8	0,05	3	2,95	$6,64.10^{-4}$
1+9	0,05	3,5	3,45	$7,76.10^{-4}$
1+1	0,05	4	3,95	$8,89.10^{-4}$
1+10	0,05	4,5	4,45	$1,00.10^{-3}$
1+11	0,05	5	4,95	$1,11.10^{-3}$
1+12	0,05	5,5	5,45	$1,23.10^{-3}$

On utilise un tableau de proportionnalité pour obtenir la mesure en mètre de la largeur de la fente. On sait pour cela que 80 pixels correspondent à 1,80 cm à l'impression. On peut ainsi passer de la mesure de la largeur de fente en pixel à une mesure en mètre.

On place ensuite chacune de nos grilles périodiques 1D dans notre montage, et après passage à la diffraction, on procède à la mesure de l'interfrange directement sur l'écran.

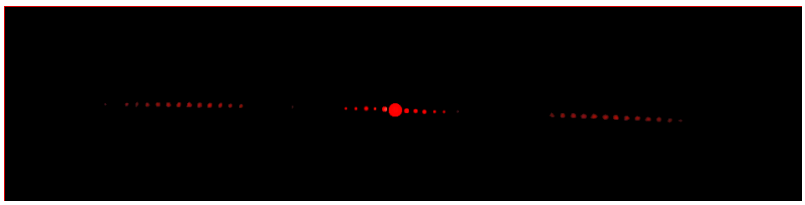


FIG. 16 – **Photo d'une figure de diffraction obtenue après passage à la diffraction de la grille décrite ci-dessus.**

Les mesures obtenues sont répertoriées dans le tableau suivant :

	Largeur des fentes (en m)	(Largeur des fentes)$^{-1}$ (en m^{-1})	Interfrange (en m)
1+4	$2,14.10^{-4}$	$4,68.10^{3}$	$2,76.10^{-2}$
1+5	$3,26.10^{-4}$	$3,07.10^{3}$	$1,86.10^{-2}$
1+6	$4,39.10^{-4}$	$2,28.10^{3}$	$1,42.10^{-2}$
1+7	$5,51.10^{-4}$	$1,81.10^{3}$	$1.13.10^{-2}$
1+8	$6,64.10^{-4}$	$1,51.10^{3}$	$9,43.10^{-3}$
1+9	$7,76.10^{-4}$	$1,29.10^{3}$	$8,00.10^{-3}$
1+1	$8,89.10^{-4}$	$1,13.10^{3}$	$7,00.10^{-3}$
1+10	$1,00.10^{-3}$	$9,99.10^{2}$	$6,25.10^{-3}$
1+11	$1,11.10^{-3}$	$8,98.10^{2}$	$5,57.10^{-3}$
1+12	$1,23.10^{-3}$	$8,15.10^{2}$	$5,13.10^{-3}$

A l'aide d'une régression linéaire (figure 17) de l'interfrange en fonction de l'inverse de la largeur de fente (1/a dans la formule ci-après), on obtient l'équation :

$$i = 3,77214.10^{-4} + \frac{5,92508.10^{-6}}{a}$$

avec un coefficient de corrélation : R = 0,999823

On a également la relation entre l'interfrange et la largeur de la fente :

$$\boxed{i = \frac{\lambda D}{a}}$$

On remarque ici que la droite expérimentale ne passe pas par l'origine alors qu'elle le devrait théoriquement. A l'aide du logiciel KaleidaGraph, on peut alors forcer la courbe à passer par l'origine, en utilisant l'outil Curve Fit Options et en sélectionnant la coche Force Line fit through zero.

On obtient alors l'équation :

$$\boxed{i = \frac{6,07344.10^{-6}}{a}}$$

avec un coefficient de corrélation : R = 0,999748

N.B. : A l'aide d'une modélisation en curve fit de l'interfrange en fonction de l'inverse de la largeur de fente, on obtenait directement l'équation ci-dessus.

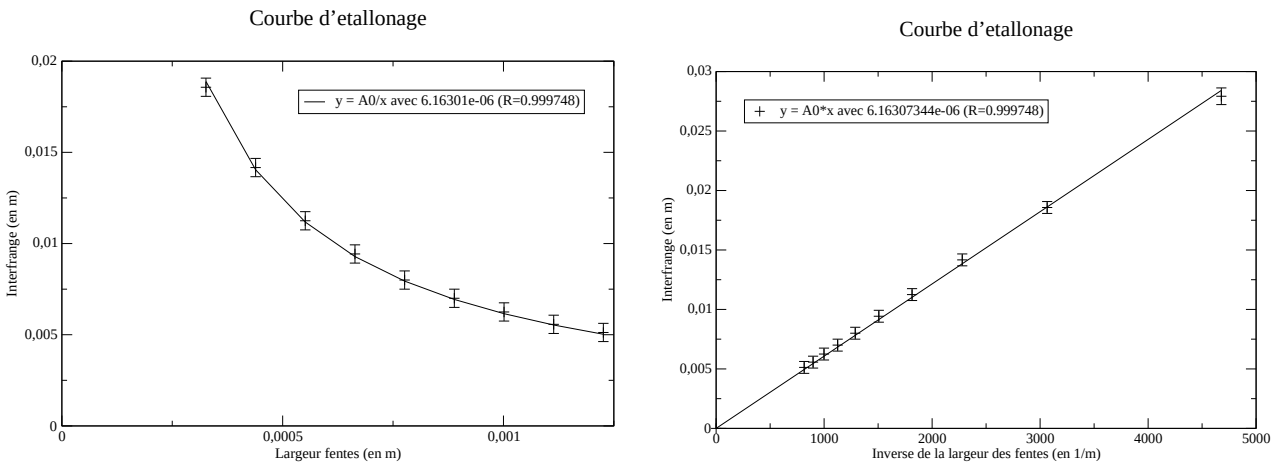


FIG. 17 – **Courbes d'étalonnage obtenues par modélisation en curve fit.**

On a la relation entre l'interfrange et la largeur de la fente :

$$i = \frac{\lambda D}{a}$$

avec $\lambda = 835nm$ longueur d'onde du laser et D distance entre l'objet et sa figure de diffraction.

Soit :

$$\lambda D = 6,07344.10^{-6}$$
$$D = \frac{6,07344.10^{-6}}{835.10^{-9}}$$
$$\boxed{D = 7,27m}$$

4 Etude théorique de la diffraction et analyse des résultats

4.1 Fentes de Young

Considérons une onde plane monochromatique, de longueur d'onde λ_0 issue d'une source S. Notons A_0 l'amplitude de l'onde émise par S. Nous allons étudier le phénomène d'interférences qui se produit lorsque le faisceau lumineux rencontre deux fentes, les fentes de Young.

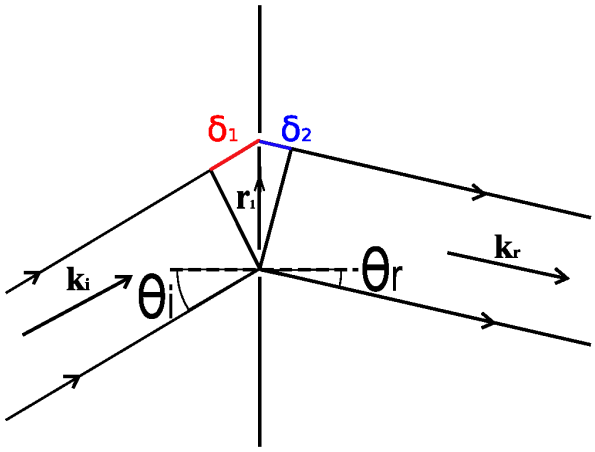


FIG. 18 – **Diffraction par les fentes de Young.** (Notons que r_n est un vecteur qui définit la deuxième fente à partir de la première.)

Dans un premier temps, calculons la différence de marche δ en utilisant les notations de la figure 18 :

$$\begin{aligned}
\delta &= \delta_1 + \delta_2 \\
\delta &= \frac{1}{||\overrightarrow{k_r}||}\overrightarrow{k_r}.\overrightarrow{r} - \frac{1}{||\overrightarrow{k_r}||}\overrightarrow{k_i}.\overrightarrow{r} \\
\delta &= \frac{\lambda_0}{2\pi}\overrightarrow{k_r}.\overrightarrow{r} - \frac{\lambda_0}{2\pi}\overrightarrow{k_i}.\overrightarrow{r} \\
\delta &= \frac{\lambda_0}{2\pi}\left(\overrightarrow{k_r} - \overrightarrow{k_i}\right).\overrightarrow{r}
\end{aligned}$$

Si on appelle $\overrightarrow{q} = \overrightarrow{k_r} - \overrightarrow{k_i}$, le vecteur diffusion, on obtient :

$$\delta = \frac{\lambda_0}{2\pi}\overrightarrow{q}.\overrightarrow{r}$$

Ainsi, comme l'onde supérieure a un déphasage de $\frac{2\pi}{\lambda_0}\delta$ par rapport à l'autre onde, on peut écrire que l'amplitude $A(\overrightarrow{q})$ en fonction de $\overrightarrow{q}$ est :

$$\begin{aligned}
A(\overrightarrow{q}) &= A_0 + A_0 e^{i\frac{2\pi}{\lambda_0}\delta} \\
A(\overrightarrow{q}) &= A_0 + A_0 e^{i\overrightarrow{q}.\overrightarrow{r}}
\end{aligned}$$

$$\boxed{A(\overrightarrow{q}) = A_0(1 + e^{i\overrightarrow{q}.\overrightarrow{r}})}$$

Il faut maintenant calculer l'intensité lumineuse $I = |A|^2$ car c'est ce qui est observable sur l'écran (on pose $I_0 = (A_0)^2$) :

$$\begin{aligned}
I(\overrightarrow{q}) &= |A(\overrightarrow{q})|^2 = A(\overrightarrow{q})A^*(\overrightarrow{q}) \\
I(\overrightarrow{q}) &= A_0(1 + e^{i\overrightarrow{q}.\overrightarrow{r}})A_0(1 + e^{-i\overrightarrow{q}.\overrightarrow{r}}) \\
I(\overrightarrow{q}) &= I_0(2 + e^{i\overrightarrow{q}.\overrightarrow{r}} + e^{-i\overrightarrow{q}.\overrightarrow{r}}) \\
I(\overrightarrow{q}) &= I_0(2 + 2\frac{e^{i\overrightarrow{q}.\overrightarrow{r}} + e^{-i\overrightarrow{q}.\overrightarrow{r}}}{2})
\end{aligned}$$

$$\boxed{I(\overrightarrow{q}) = 2I_0(1 + cos(\overrightarrow{q}.\overrightarrow{r}))}$$

4.2 Diffraction par N fentes linéaires

4.2.1 Cas général

Considérons maintenant N fentes $F_1, F_2, \ldots, F_N$ disposés comme sur la figure 19 :

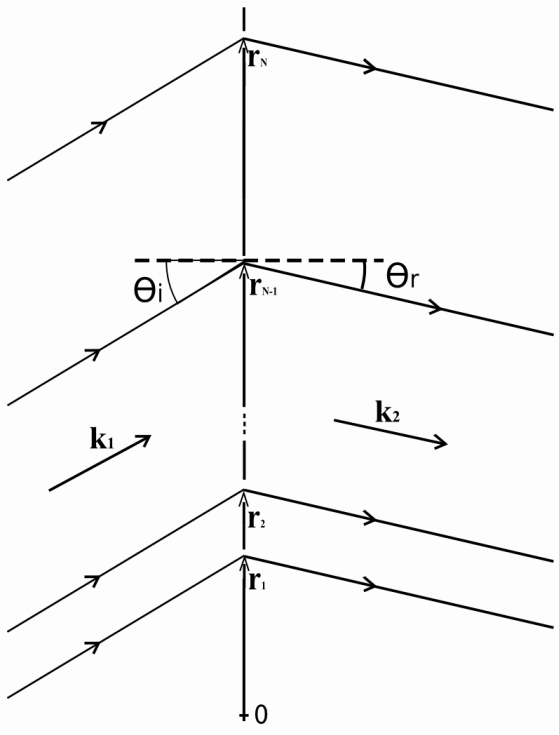


FIG. 19 – **Diffraction par N fentes**

En utilisant les notions et les notations du paragraphe précédent, on trouve que l'onde issue de la fente F_n ($n \in \{1, 2, \ldots, N\}$) a un déphasage par rapport à l'onde issue de F_1 :

$$\frac{2\pi}{\lambda_0}\delta = \overrightarrow{q}.\overrightarrow{r_n}$$

Ainsi, on obtient :

$$\begin{aligned} A(\overrightarrow{q}) &= A_0(e^{i\frac{2\pi}{\lambda_0}\delta_1} + A_0 e^{i\frac{2\pi}{\lambda_0}\delta_2} + \ldots + A_0 e^{i\frac{2\pi}{\lambda_0}\delta_N}) \\ A(\overrightarrow{q}) &= A_0(e^{i\frac{2\pi}{\lambda_0}\delta_1} + e^{i\frac{2\pi}{\lambda_0}\delta_2} + \ldots + e^{i\frac{2\pi}{\lambda_0}\delta_N}) \\ A(\overrightarrow{q}) &= A_0(e^{i\overrightarrow{q}.\overrightarrow{r_1}} + e^{i\overrightarrow{q}.\overrightarrow{r_2}} + \ldots + e^{i\overrightarrow{q}.\overrightarrow{r_N}}) \end{aligned}$$

On obtient donc la formule

$$\boxed{A(\overrightarrow{q}) = A_0 \sum_{n=1}^{N} e^{i\overrightarrow{q}.\overrightarrow{r_n}}} \quad (3)$$

Cette formule ne peut pas être simplifiée dans le cadre général. C'est pour cela que nous allons étudier des cas particuliers

4.2.2 Fentes équidistantes

Considérons maintenant N fentes équidistantes et alignées. On a : $\overrightarrow{r_n} = n\overrightarrow{r_1}$. Posons $\overrightarrow{r} = \overrightarrow{r_1}$. On utilise les notations précédentes et celles de la figure 20.

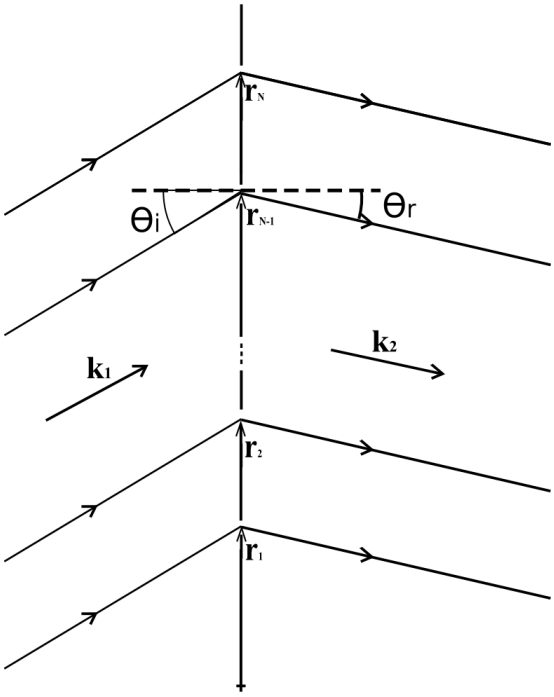


FIG. 20 – **Diffraction par N fentes équidistantes**

L'utilisation de la formule 3 du cas général donne ici :

$$\begin{aligned} A(\overrightarrow{q}) &= A_0 \textstyle\sum_{n=1}^{N} e^{i\overrightarrow{q}.\overrightarrow{r_n}} \\ A(\overrightarrow{q}) &= A_0 \textstyle\sum_{n=1}^{N} e^{i.n.\overrightarrow{q}.\overrightarrow{r}} \\ A(\overrightarrow{q}) &= A_0 \textstyle\sum_{n=1}^{N} \left(e^{i\overrightarrow{q}.\overrightarrow{r}}\right)^n \end{aligned}$$

On obtient donc :

$$\boxed{A(\overrightarrow{q}) = A_0 \frac{1 - e^{iN\overrightarrow{q}.\overrightarrow{r}}}{1 - e^{i\overrightarrow{q}.\overrightarrow{r}}}} \tag{4}$$

Nous allons maintenant calculer l'intensité de l'onde en fonction de $\overrightarrow{q}$:

$$I(\overrightarrow{q}) = |A(\overrightarrow{q})|^2$$

$$I(\overrightarrow{q}) = \left| A_0^2 e^{i\frac{N-1}{2}\overrightarrow{q}.\overrightarrow{r}} \frac{\sin^2(\frac{N}{2}\overrightarrow{q}.\overrightarrow{r})}{\sin^2(\frac{1}{2}\overrightarrow{q}.\overrightarrow{r})} \right|$$

Par conséquent en posant $I_0 = A_0^2$, cette expression nous donne :

$$\boxed{I(\overrightarrow{q}) = I_0 \frac{\sin^2\left(\frac{N}{2}\overrightarrow{q}.\overrightarrow{r}\right)}{\sin^2\left(\frac{1}{2}\overrightarrow{q}.\overrightarrow{r}\right)}} \tag{5}$$

On a aussi en utilisant les angles de la figure 20 :

$$I(\theta_i, \theta_r) = I_0 \frac{\sin^2\left(\frac{N}{2}\frac{2\pi}{\lambda}r(\sin(\theta_r) - \sin(\theta_i))\right)}{\sin^2\left(\frac{1}{2}\frac{2\pi}{\lambda}r(\sin(\theta_r) - \sin(\theta_i))\right)}$$

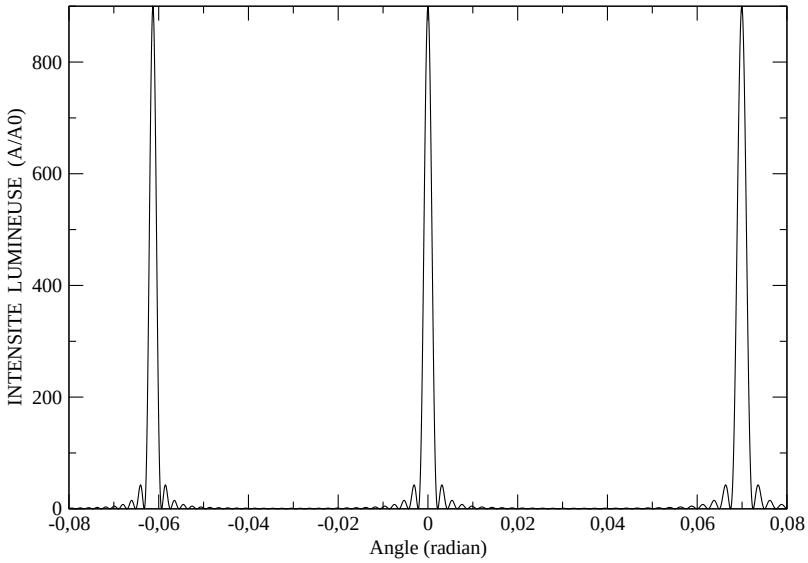


FIG. 21 – **Simulation de la diffraction d'un faisceau d'ondes monochromatiques (λ=650nm) par N=30 fentes équidistantes de distance 50μm et où $\theta_i = 0$. Le pas angulaire de la simulation est 10^5 rad.** (La simulation a été réalisée par un programme écrit en C (Annexe C) et la courbe a été représentée par le logiciel xmgrace$^{\star 2}$)

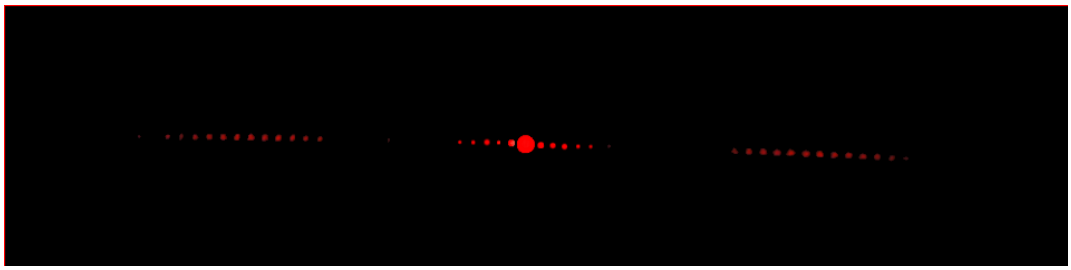


FIG. 22 – **Photo de la figure de diffraction d'une grille de lignes verticales et régulières. On voit que cela correspond bien au modèle théorique. La photo a été retouchée informatiquement.**

4.2.3 Pertubation sinusoïdale, approximation des trois tâches

Afin de complexifier la situation dans le but de mieux comprendre par la suite les figures de diffraction plus compliquées, nous allons maintenant prendre des fentes équidistantes et les déplacer, les écarter ou les rapprocher entre elles.

De façon plus précise, considérons les notations et le réseau de fentes régulières du paragraphe précédent. Ainsi, la fente n se situe à la position (r est la distance entre deux fentes) :

$$\overrightarrow{r_n} = n.\ \overrightarrow{r}$$

Déplaçons cette fente, dans l'axe des fentes, d'une distance :

$$\epsilon.\sin(\frac{2\pi}{\lambda}n)$$

où ϵ caractérise l'amplitude de la perturbation que l'on suppose faible. La constante λ caractérise la longueur d'onde de la perturbation. Elle peut être soit un rationnel, soit un irrationnel.

Finalement, la position de la fente n est :

$$\boxed{\underline{\overrightarrow{r_n}} = \left(n.r + \epsilon.\sin(\frac{2\pi}{\lambda}.n)\right).\ \overrightarrow{e_r}}$$

Ce qui donne avec les notations complexes :

$$\overrightarrow{r_n} = \left(n.r + \epsilon.e^{i.\frac{2\pi}{\lambda}.n}\right).\ \overrightarrow{e_r}$$

La formule 3 pour le cas général devient donc :

$$\begin{aligned}
A(\overrightarrow{q}) &= A_0 \textstyle\sum_{n=1}^{N} e^{i.n.r\,\overrightarrow{q}.\overrightarrow{e_r}} \\
A(\overrightarrow{q}) &= A_0 \textstyle\sum_{n=1}^{N} e^{i\left(n.r+\epsilon.e^{i.\frac{2\pi}{\lambda}.n}\right).\overrightarrow{q}.\overrightarrow{e_r}} \\
A(\overrightarrow{q}) &= A_0 \textstyle\sum_{n=1}^{N} e^{i.n.\overrightarrow{q}.\overrightarrow{r}} e^{i.\overrightarrow{q}.\overrightarrow{e_r}.\epsilon.e^{i.\frac{2\pi}{\lambda}.n}}
\end{aligned}$$

On fait l'hypothèse que $|\overrightarrow{q}.\overrightarrow{e_r}.\epsilon.e^{i.\frac{2\pi}{\lambda}.n}| = \epsilon.\overrightarrow{q}.\overrightarrow{r} << 1$. On appellera cela par la suite l'«**approximation des trois tâches**».

Lorsque l'on fait une mesure, ϵ est fixé et seulement $\overrightarrow{q}.\overrightarrow{r}$ varie. Ainsi, cette approximation n'est valable que si $\overrightarrow{q}.\overrightarrow{r} << \frac{1}{\epsilon}$. Ainsi, on pourra seulement étudier la fonction au voisinage de la tâche centrale. Et plus ϵ (donc l'amplitude de la perturbation) sera petite plus le domaine de validité de l'approximation sera grand.

Cette approximation nous permet de faire un développement limité et nous permettra d'avoir une expression mathématique de la situation au voisinage de la tache centrale.

$$\begin{aligned}
A(\overrightarrow{q}) &= A_0 \textstyle\sum_{n=1}^{N} e^{i.n.\overrightarrow{q}.\overrightarrow{r}}(1+\epsilon.\overrightarrow{q}.\overrightarrow{e_r}.e^{i.\frac{2\pi}{\lambda}.n}) \\
A(\overrightarrow{q}) &= A_0 \textstyle\sum_{n=1}^{N} e^{i.n.\overrightarrow{q}.\overrightarrow{r}} + A_0.\epsilon.\overrightarrow{q}.\overrightarrow{e_r}.\sum_{n=1}^{N} e^{i.\left(\frac{2\pi}{\lambda}+\overrightarrow{q}.\overrightarrow{r}\right).n} \\
A(\overrightarrow{q}) &= A_r(\overrightarrow{q}) + \epsilon.A_p(\overrightarrow{q})
\end{aligned}$$

avec $A_r(\overrightarrow{q}) = A_0 \sum_{n=1}^{N} e^{i.n.\overrightarrow{q}.\overrightarrow{r}}$ et $A_p(\overrightarrow{q}) = A_0.\overrightarrow{q}.\overrightarrow{e_r}.\sum_{n=1}^{N} e^{i.\left(\frac{2\pi}{\lambda}+\overrightarrow{q}.\overrightarrow{r}\right).n}$

En faisant les mêmes calculs que pour obtenir les équations 4 et 5 nous obtenons :

$$\begin{aligned}
|A_r(\overrightarrow{q})| &= |A_0 \tfrac{1-e^{iN\,\overrightarrow{q}.\overrightarrow{r}}}{1-e^{i\,\overrightarrow{q}.\overrightarrow{r}}}| \\
|A_r(\overrightarrow{q})| &= |A_0 e^{i\frac{N-1}{2}\overrightarrow{q}.\overrightarrow{r}} \tfrac{\sin\left(\frac{N}{2}\overrightarrow{q}.\overrightarrow{r}\right)}{\sin\left(\frac{1}{2}\overrightarrow{q}.\overrightarrow{r}\right)}| \\
|A_r(\overrightarrow{q})| &= A_0 \left|\tfrac{\sin\left(\frac{N}{2}\overrightarrow{q}.\overrightarrow{r}\right)}{\sin\left(\frac{1}{2}\overrightarrow{q}.\overrightarrow{r}\right)}\right|
\end{aligned}$$

alors

$$I_r(\overrightarrow{q}) = I_0 \frac{\sin^2\left(\frac{N}{2}\overrightarrow{q}.\overrightarrow{r}\right)}{\sin^2\left(\frac{1}{2}\overrightarrow{q}.\overrightarrow{r}\right)}$$

Le calcul de $A_p(\overrightarrow{q})$ donne :

$$\begin{aligned}
|A_p(\overrightarrow{q})| &= |A_0.\overrightarrow{q}.\overrightarrow{e_r}.\tfrac{1-e^{i.N.\left(\frac{2\pi}{\lambda}+\overrightarrow{q}.\overrightarrow{r}\right)}}{1-e^{i.\left(\frac{2\pi}{\lambda}+\overrightarrow{q}.\overrightarrow{r}\right)}}| \\
|A_p(\overrightarrow{q})| &= |A_0.\overrightarrow{q}.\overrightarrow{r}.e^{i\frac{N-1}{2}\left(\frac{2\pi}{\lambda}+\overrightarrow{q}.\overrightarrow{r}\right)}\tfrac{\sin\left(\frac{N}{2}\left(\frac{2\pi}{\lambda}+\overrightarrow{q}.\overrightarrow{r}\right)\right)}{\sin\left(\frac{2}{2}\left(\frac{2\pi}{\lambda}+\overrightarrow{q}.\overrightarrow{r}\right)\right)}| \\
|A_p(\overrightarrow{q})| &= \overrightarrow{q}.\overrightarrow{r}.A_0.\left|\tfrac{\sin\left(\frac{N}{2}\left(\frac{2\pi}{\lambda}+\overrightarrow{q}.\overrightarrow{r}\right)\right)}{\sin\left(\frac{1}{2}\left(\frac{2\pi}{\lambda}+\overrightarrow{q}.\overrightarrow{r}\right)\right)}\right|
\end{aligned}$$

On peut donc remarquer que :

$$|A_p(\overrightarrow{q})| = \overrightarrow{q}.\overrightarrow{r}.\left|A_r\left(\overrightarrow{q} + \frac{2\pi}{\lambda}\frac{\overrightarrow{r}}{||\overrightarrow{r}||}\right)\right|$$

Ou alors par symétrie du système :

$$|A_p(\overrightarrow{q})| = \overrightarrow{q}.\overrightarrow{r}.\left|A_r\left(\overrightarrow{q} - \frac{2\pi}{\lambda}\frac{\overrightarrow{r}}{||\overrightarrow{r}||}\right)\right|$$

Ainsi, pour ce qui concerne l'intensité lumineuse, on a :

$$\begin{array}{rcl} I(\overrightarrow{q}) & = & A(\overrightarrow{q})^2 \quad = \quad (A_r(\overrightarrow{q}) + \epsilon.A_p(\overrightarrow{q}))^2 \\ I(\overrightarrow{q}) & = & (A_r(\overrightarrow{q}))^2 + \epsilon^2.(A_p(\overrightarrow{q}))^2 + 2.\epsilon\,(A_r(\overrightarrow{q}))\,(A_p(\overrightarrow{q}))^2 \\ I(\overrightarrow{q}) & = & I_r(\overrightarrow{q}) + \epsilon^2\,(A_p(\overrightarrow{q}))^2 + 2.\epsilon.A_r(\overrightarrow{q})A_p(\overrightarrow{q}) \end{array}$$

Mais comme $A_r(\overrightarrow{q})$ et $A_p(\overrightarrow{q}) = \overrightarrow{q}.\overrightarrow{r}.\left|A_r\left(\frac{2\pi}{\lambda}\frac{\overrightarrow{r}}{||\overrightarrow{r}||} \pm \overrightarrow{q}\right)\right|$ sont nuls presque partout, et que leurs maxima ne sont pas au même endroit, alors on en déduit que $A_r(\overrightarrow{q}).A_p(\overrightarrow{q}) = O \ \ \forall\overrightarrow{q}$. On a donc :

$$I(\overrightarrow{q}) \quad = \quad I_r(\overrightarrow{q}) + \epsilon^2 I_p(\overrightarrow{q})$$

$$\boxed{I(\overrightarrow{q}) \quad = \quad I_r(\overrightarrow{q}) + \epsilon^2.\overrightarrow{q}.\overrightarrow{r}.\left|A_r\left(\frac{2\pi}{\lambda}\frac{\overrightarrow{r}}{||\overrightarrow{r}||} \pm \overrightarrow{q}\right)\right|} \tag{6}$$

Grâce à cette formule, on voit qu'il va y avoir les mêmes tâches que s'il n'y avait pas la perturbation (tâches principales) et des tâches satellites autour de chacune des tâches «normales» (sauf autour de la tache centrale car pour cette tache, $\overrightarrow{q}.\overrightarrow{r} = 0$). Il ne faut pas non plus oublier l'approximation que l'on a faite pour obtenir cette formule : cela n'est vrai que si l'on est proche de la tâche centrale. C'est l'«approximation des trois tâches».

De plus, dans le cadre de cette approximation, on peut dire que la distance entre les deux satellites est $2.\frac{2\pi}{\lambda} = \frac{4\pi}{\lambda}$

Par conséquent, la mesure de cette distance permet de retrouver assez facilement la longueur d'onde λ de la perturbation sinusoïdale.

Remarque : Cela correspond bien au cas régulier de la section précédente car si $\epsilon = 0$ (pas de perturbation), alors d'une part l'approximation des trois tâches est vraie partout et en plus on voit dans 6 que le terme perturbatif n'aura plus d'intensité lumineuse : il n'y aura pas les satellites. Si maintenant λ tend vers l'infini (perturbation invisible), alors la distance entre les deux satellites va devenir nulle : ils seront sur une tâche principale (cette approximation n'est vraie que si l'approximation des trois tâches est vérifiée).

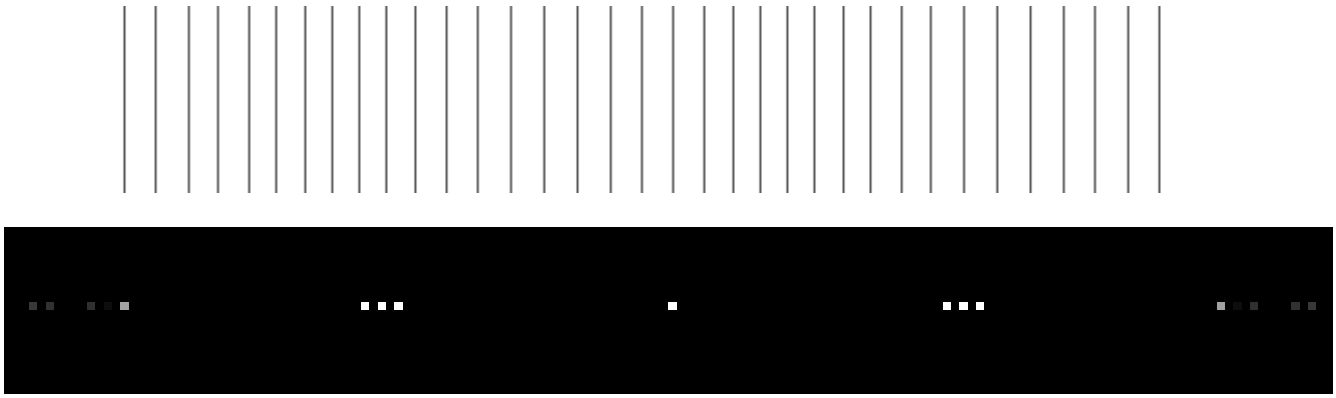


FIG. 23 – **Image de lignes ayant une perturbation sinusoïdale et sa figure de diffraction.**
Figure supérieur : extrait de lignes ayant une perturbation sinusoïdale. L'image entière est au format PostScript. La méthode de réalisation est expliquée plus loin. Il est difficile de voir la perturbation car elle est de faible amplitude.
Figure inférieur : Figure de diffraction obtenue grâce au logiciel ImageJ qui a fait la transformée de Fourier de l'image entière. Les tâche sur le coté de la figure (plus faible intensité) montrent les limites de l'approximation des trois tâche.

4.3 Diffraction par N trous dans un plan

Nous venons d'étudier la diffraction par des fentes. Nous allons maintenant étudier la diffraction par des trous (ou par des points, le résultat sera le même).

4.3.1 Cas général

Considérons maintenant les N trous disposés sur un plan comme sur la figure 24.

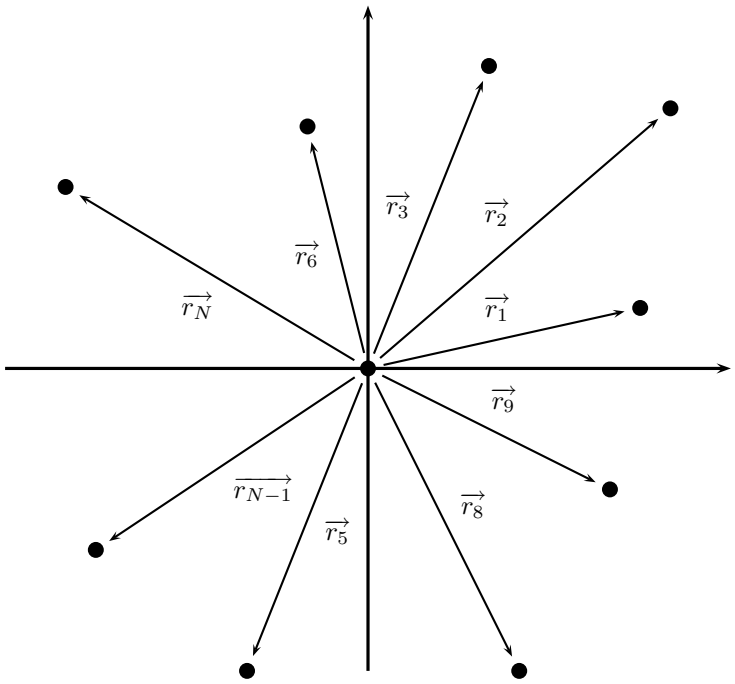


FIG. 24 – N **points sur un plan**

On remarque que la formule que l'on avait obtenue pour N fentes s'adapte bien lorsque l'on a des trous sur un plan. On a donc :

$$\boxed{A(\overrightarrow{q}) \;=\; A_0 \sum_{n=1}^{N} e^{i\overrightarrow{q}.\overrightarrow{r_n}}}$$

La différence avec le cas linéaire est que les vecteurs $\overrightarrow{k_i}$ et $\overrightarrow{k_r}$ ne sont plus nécessairement sur un même plan.

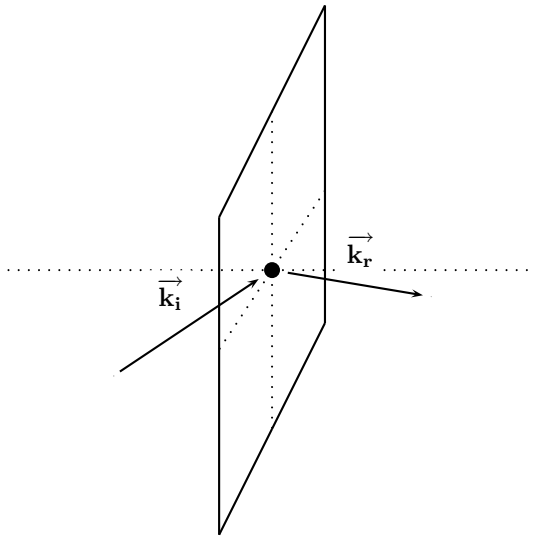


FIG. 25 – **Représentation des vecteurs $\mathbf{k_i}$ et $\mathbf{k_r}$ dans l'espace. Le plan représenté est le plan des trous.**

4.3.2 Trous régulièrement répartis

Considérons maintenant que les trous sont répartis aux intersections d'une grille plane régulière. Nous utiliserons par la suite les notations de la figure 26.

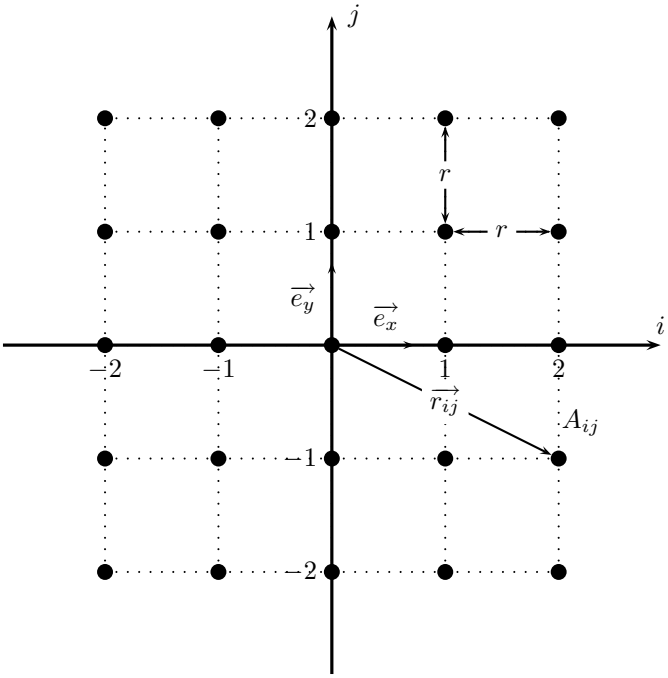


FIG. 26 – **Points sur une grille plane régulière**

On remarque que $\overrightarrow{r_{ij}} = r.(i.\overrightarrow{e_x} + j.\overrightarrow{e_y})$. Ainsi, on a :

$$\begin{aligned} A(\overrightarrow{q}) &= A_0 \sum_i^N \sum_j^N e^{\mathbf{i}\overrightarrow{q}.\overrightarrow{r_{ij}}} \\ A(\overrightarrow{q}) &= A_0 \sum_i^N \sum_j^N e^{\mathbf{i}.r.\left(i.\overrightarrow{q}.\overrightarrow{e_x}+j.\overrightarrow{q}.\overrightarrow{e_y}\right)} \\ A(\overrightarrow{q}) &= A_0 \left(\sum_i^N \left((e^{\mathbf{i}.r.\overrightarrow{q}.\overrightarrow{e_x}} \right)^i + \sum_j^N \left(e^{\mathbf{i}.r.\overrightarrow{q}.\overrightarrow{e_y}} \right)^j \right) \end{aligned}$$

On obtient donc :

$$\boxed{A(\overrightarrow{q}) = A_0 \left(\frac{1 - e^{\mathbf{i}N.r.\overrightarrow{q}.\overrightarrow{e_x}}}{1 - e^{\mathbf{i}.r.\overrightarrow{q}.\overrightarrow{e_x}}} + \frac{1 - e^{\mathbf{i}N.r.\overrightarrow{q}.\overrightarrow{e_y}}}{1 - e^{\mathbf{i}.r.\overrightarrow{q}.\overrightarrow{e_y}}} \right)}$$

On remarque que $A(\overrightarrow{q})$ est de la forme :

$$A(\overrightarrow{q}) = A_x(\overrightarrow{q}) + A_y(\overrightarrow{q})$$

où $A_x(\overrightarrow{q})$ est sa composante sur x et $A_y(\overrightarrow{q})$ est sa composante sur y.

Calculons maintenant l'intensité lumineuse : (avec $I_0 = A_0^2$)

$$\begin{aligned} I(\overrightarrow{q}) &= A_x^2 + A_y^2 + A_x A_y \\ I(\overrightarrow{q}) &= I_0^2 \left(\frac{\sin^2\left(\frac{N.r}{2}\overrightarrow{q}.\overrightarrow{e_x}\right)}{\sin^2\left(\frac{r}{2}\overrightarrow{q}.\overrightarrow{e_x}\right)} + \frac{\sin^2\left(\frac{N.r}{2}\overrightarrow{q}.\overrightarrow{e_y}\right)}{\sin^2\left(\frac{r}{2}\overrightarrow{q}.\overrightarrow{e_y}\right)} + \frac{1-e^{\mathbf{i}N.r.\overrightarrow{q}.\overrightarrow{e_x}}}{1-e^{\mathbf{i}.r.\overrightarrow{q}.\overrightarrow{e_x}}} \cdot \frac{1-e^{\mathbf{i}N.r.\overrightarrow{q}.\overrightarrow{e_y}}}{1-e^{\mathbf{i}.r.\overrightarrow{q}.\overrightarrow{e_y}}} \right) \\ I(\overrightarrow{q}) &= I_0^2 \left(\frac{\sin^2\left(\frac{N.r}{2}\overrightarrow{q}.\overrightarrow{e_x}\right)}{\sin^2\left(\frac{r}{2}\overrightarrow{q}.\overrightarrow{e_x}\right)} + \left| \frac{\sin^2\left(\frac{N.r}{2}\overrightarrow{q}.\overrightarrow{e_y}\right)}{\sin^2\left(\frac{r}{2}\overrightarrow{q}.\overrightarrow{e_y}\right)} \right| + \left| \frac{1-e^{\mathbf{i}N.r.\overrightarrow{q}.\overrightarrow{e_x}}}{1-e^{\mathbf{i}.r.\overrightarrow{q}.\overrightarrow{e_x}}} \right| \cdot \left| \frac{1-e^{\mathbf{i}N.r.\overrightarrow{q}.\overrightarrow{e_y}}}{1-e^{\mathbf{i}.r.\overrightarrow{q}.\overrightarrow{e_y}}} \right| \right) \end{aligned}$$

$$\text{et aussi : } I(\overrightarrow{q}) = I_x(\overrightarrow{q}) + I_y(\overrightarrow{q}) + \sqrt{I_x(\overrightarrow{q})}.\sqrt{I_y(\overrightarrow{q})}$$

Pour pouvoir afficher des courbes dépendant de moins de paramètres, on va considérer que $\overrightarrow{k_i}$ est normal au plan des points.. On est donc dans la situation de la figure 27. Les courbes obtenues devraient donc être en 3 dimensions. Cependant, afin d'une part de faciliter la compréhension de la courbe et aussi pour des raisons techniques, (à titre d'information, le fichier de sortie du programme qui génère la version en trois dimensions de la figure 28 faisait 200MO !), on se limite à des courbes à 2 dimensions.

La figure 28 est en fait une coupe horizontale de la courbe en trois dimensions. Sur cette figure, il y a que les points qui ont une intensité supérieure à une certaine valeur I_0. De plus, cette courbe est ce que l'on peut obtenir sur un écran et la valeur I_0 correspond au contraste du montage.

Les figures 29 montrent les variations de l'intensité autour de la tâche centrale. La première est une vue en coupe et la deuxième est une courbe en trois dimensions.

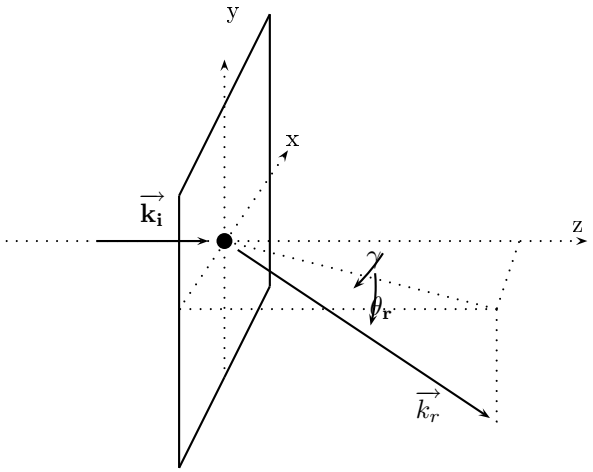


FIG. 27 – **Représentation du vecteur $\mathbf{k_i}$**

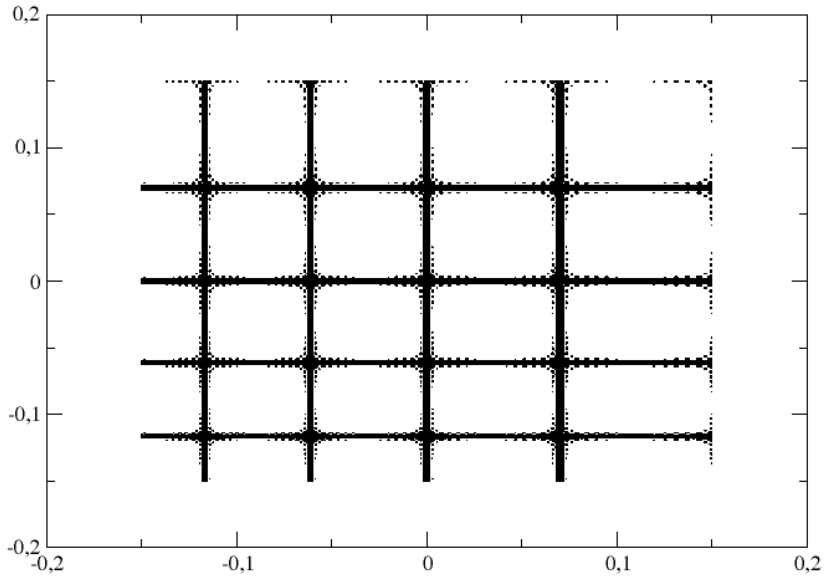


FIG. 28 – **Simulation de la diffraction d'un faisceau d'onde monochromatique (λ=650nm) par N=900 points disposés sur une grille plane et régulière. L'abcisse et l'ordonnée de la courbe sont les angles γ et θ de la figure 27.** (La simulation a été réalisée par un programme écrit en C (Annexe C et la courbe a été représentée par le logiciel xmgrace$^{\star 2}$. (linux)

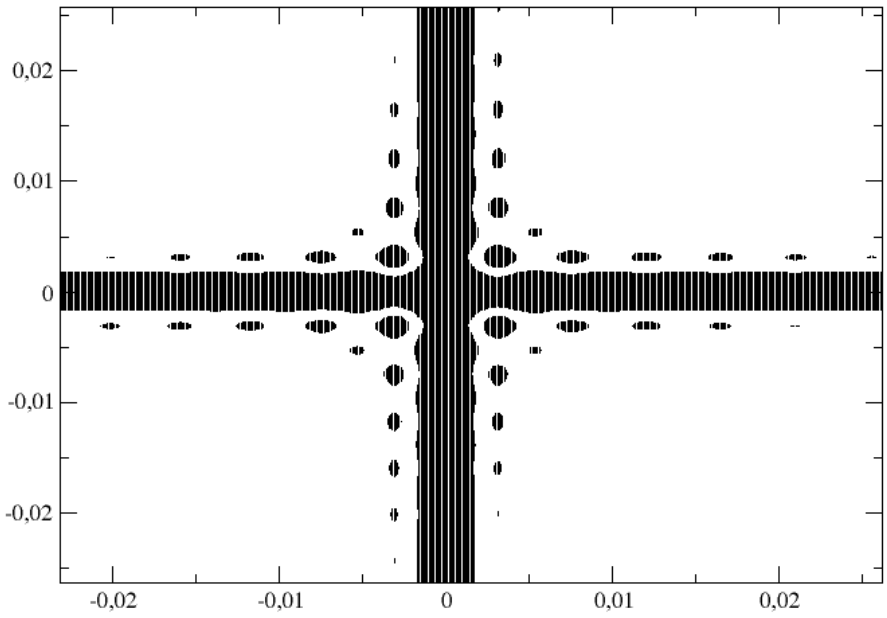


FIG. 29 – Détail de la figure 28

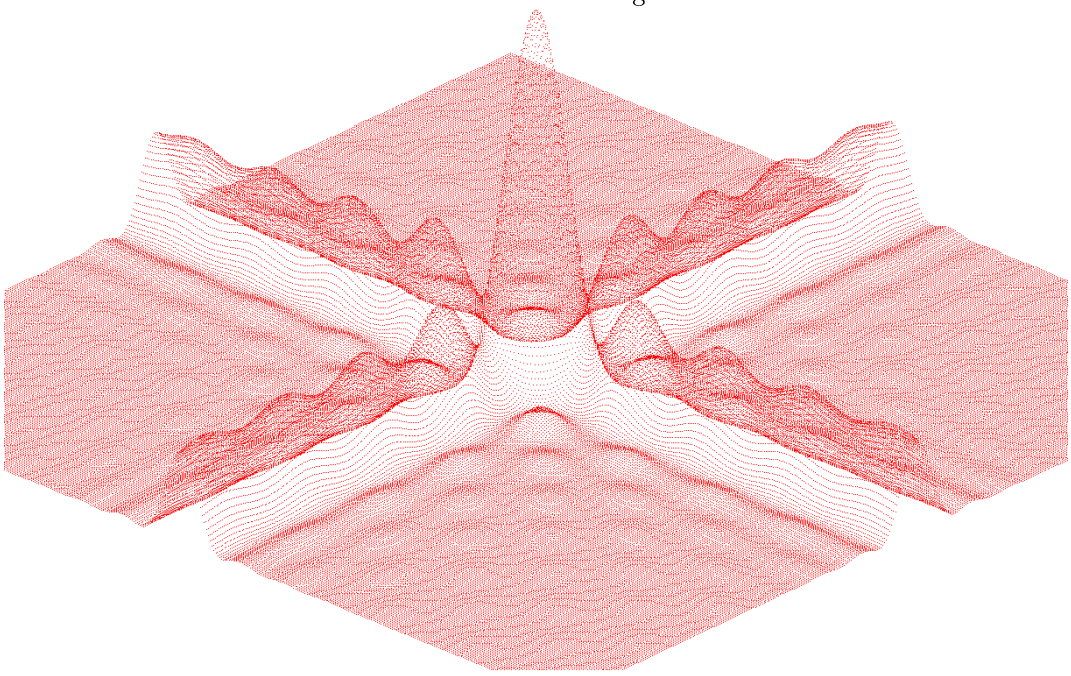


FIG. 30 – **Courbe 3D issue de la simulation de la diffraction d'un faisceau d'ondes monochromatiques (λ=650nm) par N=900 diffuseurs disposés sur une grille plane et régulière. Les axes du bas sont sont les angles γ et θ de la figure 27 et la hauteur est proportionnelle à l'intensité lumineuse. (La simulation a été réalisée par un programme écrit en C (Annexe C) et la courbe a été représentée par le logiciel gnuplot$^{\star 4}$ (linux)**

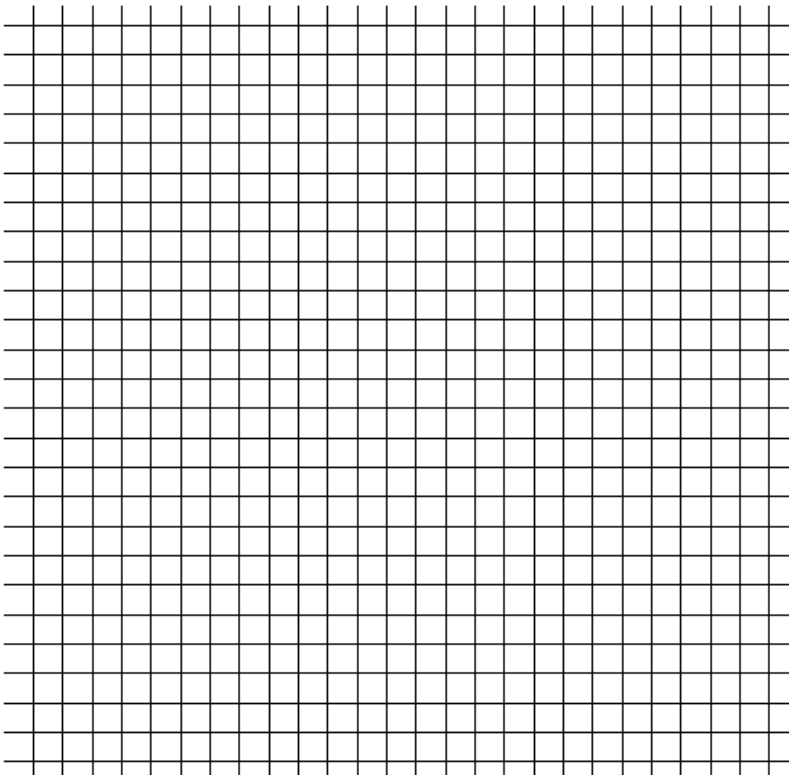


FIG. 31 – **Grille de diffraction 0°-90°**

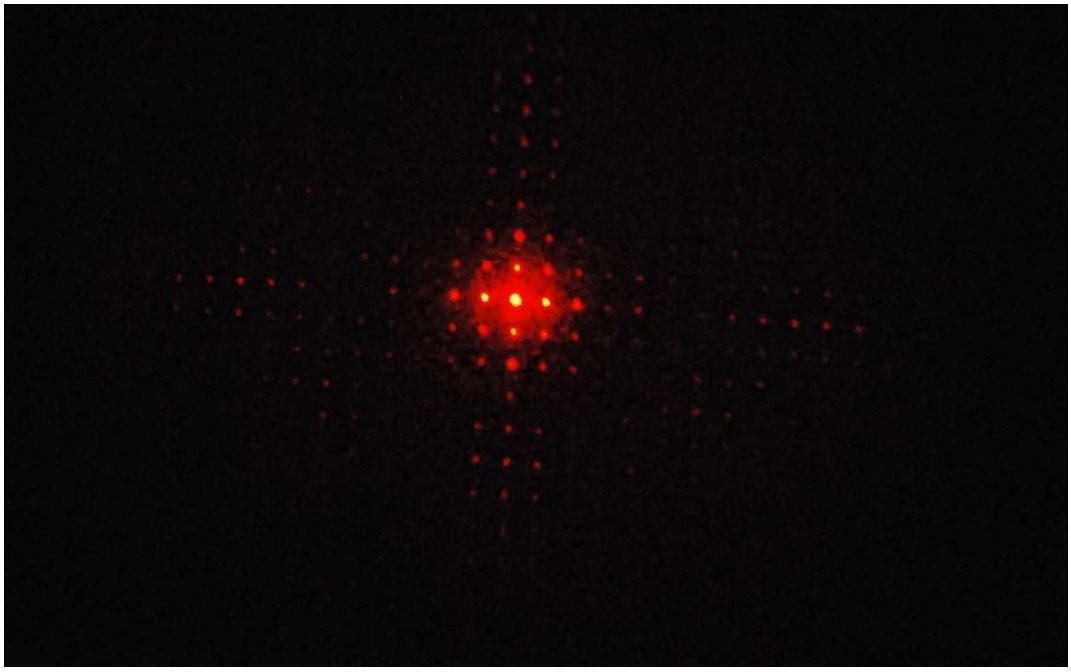


FIG. 32 – Photo de diffraction de la figure 31

4.4 Diffraction par des surfaces quasi-périodiques

Le calcul pour le cas quasi-périodique ne permet pas d'avoir une formule simplifiable. En effet, il faut faire une somme sur des termes différents qui n'ont pas spécifiquement des liens entre eux.

C'est pour cela qu'il n'y a pas de calcul dans cette section. En revanche, on peut tout de même obtenir des figures de diffraction.

La figure 33 est une photo que nous avons réalisée et retouchée par des outils informatiques. C'est la figure de diffraction des fentes de Fibonacci en une dimension.

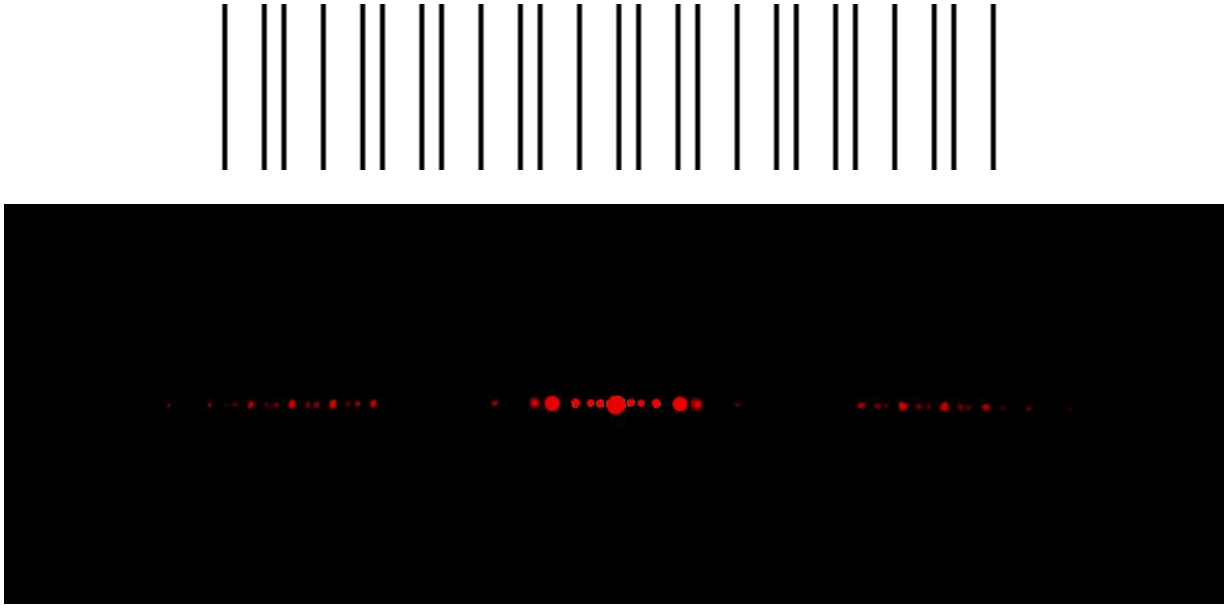
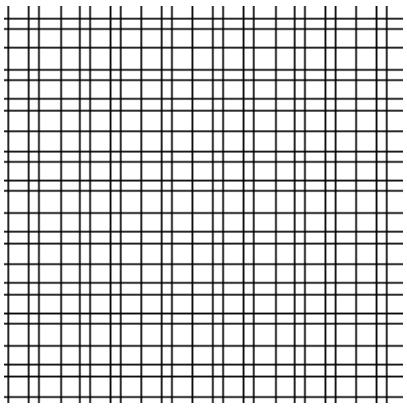


FIG. 33 – **Grille et figure de diffraction obtenue grâce à une grille de Fibonnacci. Cette photo a été améliorée grâce au logiciel ImageJ (amélioration du contraste) et du logiciel Gimp$^{\star 6}$(linux) afin d'améliorer la partie centrale**

Par contre, la figure 34 provient d'une **simulation** que nous avons obtenue grâce à un programme en C qui calcule directement la somme pour chaque direction. Un exemple de ce type de programme est en annexe.



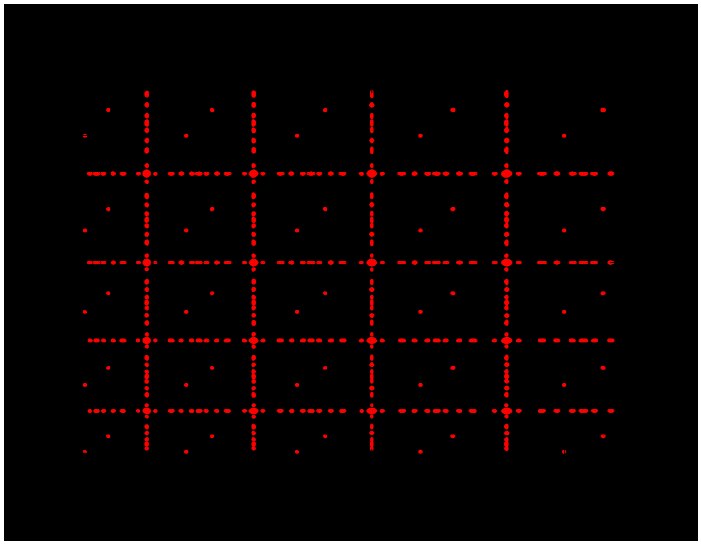


FIG. 34 –

Grille et figure de diffraction obtenue grâce à une grille de Fibonacci 0-90°. Seuls les points de forte intensité lumineuse sont représentés.

Quant à la figure 35, c'est une diffraction qui peut s'obtenir grâce au logiciel ImageJ qui fait une transformée de Fourier d'une grille comportant les 5 motifs de lignes croisées.

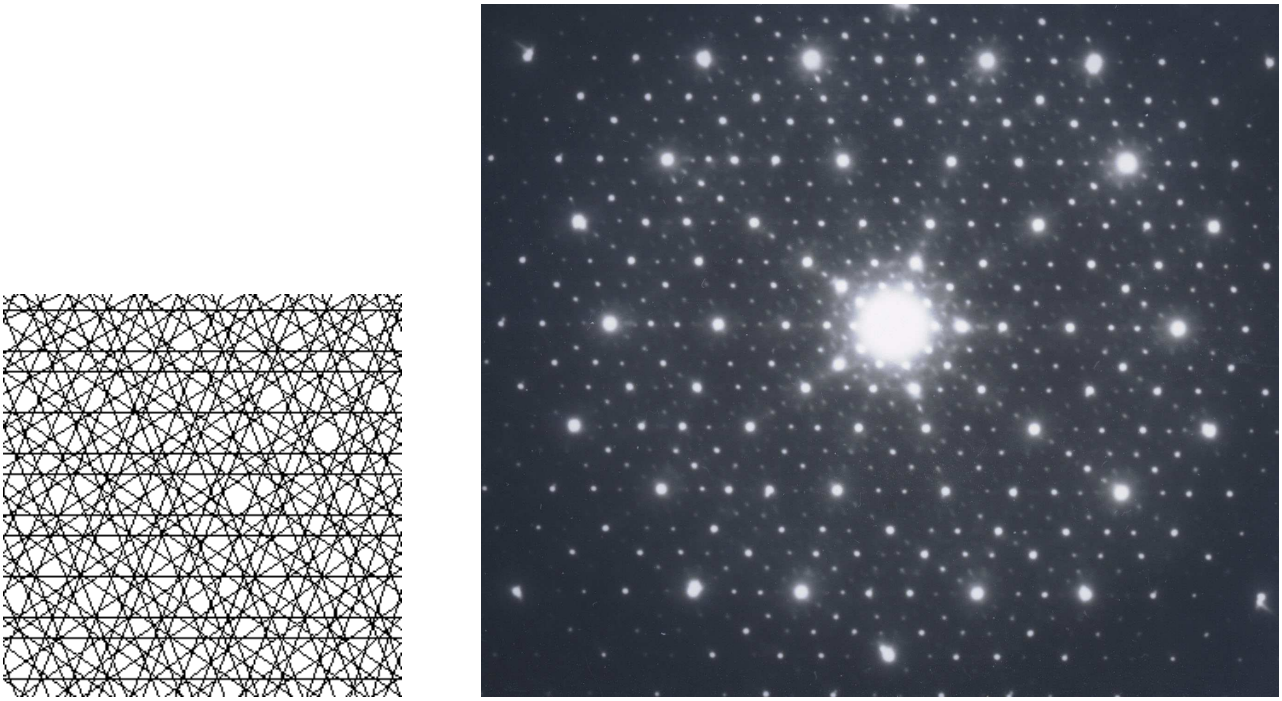


FIG. 35 – **Grille et figure de diffraction de grilles ayant un angle de 72° entre elles avec la séquence de Fibonacci**

Conséquence :
Grâce à toutes ces figures de diffraction, on remarque une caractéristique : il y a toujours deux tâches moins marquées entre deux tâches de forte intensité lumineuse. C'est une caractéristique de la diffraction des figures utilisant Fibonacci.

Conclusion

Dans le précédent dossier, nous avions tenté de comprendre la théorie de la diffraction, pour des structures périodiques, en tentant de se rapprocher le plus possible du cas quasi-périodique. Nous avions également étudié le cas des symétries autorisées et interdites dans une structure périodique. Dans cette nouvelle approche, principalement expérimentale, nous avons entrepris une expérience de diffraction optique par des surfaces périodiques et quasi-périodiques, à partir de grilles de diffraction générées par des programmes écrits en langage C. Pour ce faire, nous avons réalisé un montage optique, et envoyer un faisceau laser sur les grilles en question. Les figures de diffraction obtenues ont bien répondu à nos attentes. La relation reliant l'interfrange à la longueur d'onde et à l'écart entre les fentes a pu être retrouvée. Nous avons de même observé la périodicité des figures quand les grilles étaient périodiques, et la quasi-périodicité de celles-ci quand les grilles étaient quasi-périodiques. De plus, les symétries autorisées et interdites sont apparues dans les figures de diffraction, ce qui a confirmé la théorie.

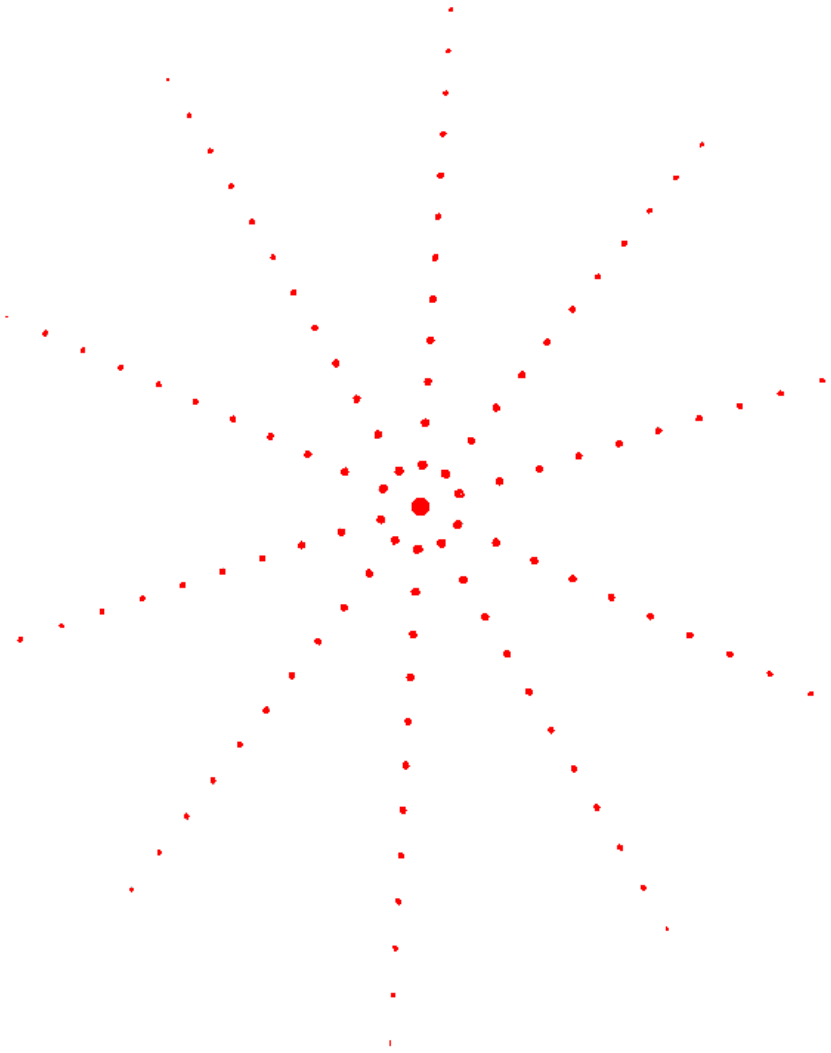


Photo d'une figure de diffraction de 5 grilles régulières ayant un angle de 72° entre elles.
Le noir a du fond a été transformé en blanc, et le contrast a été amélioré afin de faire disparaitre le bruit pour mieux voir les taches.

ANNEXES

A Bibliographie

[1] *Encyclopédie Universalis*. 2003.

[2] *Cours de physique - Ondes - BERKELEY*. DUNOD, 2004.

[3] R.D. Diehl, J. Ledieu, N. Ferralis, A.W. Szmodis, and R. McGrath. Quasicrystals : A new class of ordered structures. *Journal of Physics : Condensed Matter*, 2003.

[4] N. Ferralis, A.W. Szmodis, and R.D. Diehl. Diffraction from one- and two-dimensional quasicrystalline gratings. *American Journal of Physics*, 2004.

[5] D. Gratias. Les quasicristaux – 15 ans après, de nombreuses énigmes subsistent... *Journal of Physics – Condensed Matter*, 2003.

[6] http ://www.linuxfocus.org/Francais/May1998/article43.html. *Le Langage Postscript*.

[7] D. Levine and P.J. Steinhardt. Quasicrystals : A new class of ordered structures. *Phys. Rev.*, 1984.

[8] M. Senechal. *Quasicrystals and Geometry*. 1995.

B Logiciels utilisés

Les logiciels suivant ont permis de réaliser la plupart des figures présentes dans ce document.

★1 gcc (linux, en ligne de commande) : compilation des programmes en C

★2 xmgrace (linux) : courbes en deux dimensions

★3 Kaleidagraph (windows) : ajustement de courbe (complément de xmgrace)

★4 gnuplot (linux) : courbes en trois dimensions

★5 kghostview (linux) : affichage de fichier au format PostScript

★6 Gimp (linux) : retoucher et redimensionner les images/photos

★7 ImageJ (linux et windows) : faire des transformé de Fourier et retoucher les photos

★8 ImageMagick (linux, en ligne de commande) : conversion d'images (ps↔png↔jpg par exemple)

★9 pspicture (lignes de codes latex) : création d'images en PostScript pour les documents latex.

C Programme en C permettant de réaliser une simulation d'une diffraction par une grille d'une ou de deux dimensions

```
#include <cmath>
#include <fstream>
#include <iostream>
#include <stdio.h>
#include <string>
#include <complex>
using namespace std;
//Variables de sortie (utile pour tracer les courbes) :
ofstream sortie("sortie.dat");


main(){

//Définition des variables :
float lambda=0.00000065;
float N=30;
float thetar;
float r=0.00001;
float j;

complex<float> Intensite;
complex<int> I(0,0);
complex<float> a=(0,0);


//Début de la bouche des calculs : variation angulaire de -pi/2 à + pi/2 avec un pas de 0.00001
for (thetar=-0.08;thetar<0.08;thetar=thetar+0.000001)
{
a.real()=0;
a.imag()=0;

//Boucle pour le calcul de la somme
for (j=-N/2;j<N/2;j++)
{
float tmp;
a.real() = a.real()+cos(2*3.14159/lambda*r*sin(thetar)*j*(1-sin(thetar)));
a.imag() = a.imag()+sin(2*3.14159/lambda*r*sin(thetar)*j*(1-sin(thetar)));
nbr++;
}

Intensite=((a.real())*(a.real())+(a.imag())*(a.imag()));
cout«thetar«" "«real(Intensite)«" reel : "«real(a)«" img :"«imag(a)«endl;

sortie«thetar«" "«real(Intensite)«endl;
}
}
}
```

D Générer une suite de L et de S suivant une suite de Fibonacci

```
float x1,x2,y1,y2 ;
int fin_boucle=0 ;
int col_apr=0 ;
int tour2 ;

int mat[100][100] ;
for (int i=1 ;i<100 ;i++)
for (int j=1 ;j<100 ;j++)
mat[i][j]=5 ;
cout«i«endl ;

mat[0][0]=0 ;
for(int tour=1 ;tour<10 ;tour++)
{
col_apr=0 ;
tour2=tour ;
fin_boucle=0
for (int col_avt=0 ;fin_boucle==1 ;col_avt++)

if (mat[tour-1][col_avt]==5)
{
fin_boucle=1 ;
}

if (mat[tour-1][col_avt]==0)
{
mat[tour][col_apr]=1 ;
col_apr=col_apr+1 ;
}

if (mat[tour-1][col_avt]==1)
{
mat[tour][col_apr]=1 ;
col_apr=col_apr+1 ;
mat[tour][col_apr]=0 ;
col_apr=col_apr+1 ;
}} }

int fibo[100] ;
for (int i=1 ;i<100 ;i++)
{
fibo[i]=mat[tour2][i] ;
}
```

La séquence obtenue est une suite de 1 et de 0 cela est plus facilement utilisable que les L et les S. Le résultat est dans (fibo[i])$_i$ est écrite dans l'élément fibo.

E Programme en C permettant de générer une figure comportant 5 motifs croisées ayant un angle de 72° entre elles

```
//position dans du cadre de l'image :
posx=400;
posy=200;

//1er cas : si l'angle est négatif : ******************************************************

//max=n° ligne d'arrêt

int max;

//Cette boucle permet d'alléger le programme en faisant deux boucles. Pour la première, l'angle est -1.2 radian et -0.6 radian pour l'autre angle
for(int boucle=1;boucle<3;boucle++)
{
if(boucle==1)
{
angle=-1.2;
max=52;
}
if(boucle==2)
{
angle=-0.6;
max=56;
}

//abscisse et ordonnée du point d'intersection avec le cadre de l'image

float coo[200][4];
//i=n° de la droite
for (int i=0;i<max;i++)
{
//équation de la droite à tracer y=a1x+b1 avec :

a1=tan(angle);
b1=i/abs(cos(angle));

//Intersection avec l'axe des x : (x,y)
x=-b1/a1;//a1<0

if (x<40)
{
coo[i][0]=-b1/a1;
coo[i][2]=0;
}
else
{
coo[i][0]=a1*40+b1;
coo[i][2]=1;
}

//Intersection avec l'axe des y : (x,y)
//On met dans coo[][2] un 1 si l'on sort du cadre (0 sinon).
y=b1;
if (y<40)
{
coo[i][1]=b1;
coo[i][3]=0;
}
else
{
```

```
coo[i][1]=(40-b1)/a1 ;
coo[i][3]=1 ;
}


if(coo[i][2]==0 and coo[i][3]==0)
{
//sortie«endl«"0.1 setlinewidth"«endl;
sortie«"n"«endl;
sortie«posx+2*coo[i][0] «" "« posy «" m"«endl;
sortie«posx «" "« posy+2*coo[i][1] «" l"«endl;
sortie«"s"«endl;
}

if(coo[i][2]==1 and coo[i][3]==0)
{
//sortie«endl«"0.3 setlinewidth"«endl;
//depassement sur les x
sortie«"n"«endl;
sortie«posx+80 «" "«posy+2*coo[i][0] «" m"«endl;
sortie«posx «" "«posy+2*coo[i][1] «" l"«endl;
sortie«"s"«endl;
}

if(coo[i][2]==0 and coo[i][3]==1)
{
//sortie«endl«"0.5 setlinewidth"«endl;
//dépassement sur les y
sortie«"n"«endl;
sortie«posx+2*coo[i][0] «" "«posy «" m"«endl;
sortie«posx+2*coo[i][1] «" "«posy+80 «" l"«endl;
sortie«"s"«endl;
}

if(coo[i][2]==1 and coo[i][3]==1)
{
//sortie«endl«"0.7 setlinewidth"«endl;
sortie«"n"«endl;
sortie«posx+80 «" "«posy+2*coo[i][0] «" m"«endl;
sortie«posx+2*coo[i][1] «" "«posy+80 «" l"«endl;
sortie«"s"«endl;
}
}
}


//2eme cas : si l'angle est positif : ****************************************

for(int boucle=1;boucle<3;boucle++)
{
if(boucle==1)
{
angle=1.2 ;
max=52;
}
if(boucle==2)
{
angle=0.6 ;
max=56;
}
float coo[200][4];

coo[200][4];
for (int i=0;i<max;i++)
{

a1=tan(angle);
```

```
b1=i/cos(angle) ;

x=-b1/a1 ;
if (x>-40)
{
coo[i][0]=-b1/a1 ;
coo[i][2]=0 ;
}
else
{
coo[i][0]=-a1*40+b1 ;
coo[i][2]=1 ;
}

y=b1 ;
if (y<40)
{
coo[i][1]=b1 ;
coo[i][3]=0 ;
}
else
{
coo[i][1]=(40-b1)/a1 ;
coo[i][3]=1 ;
}

if(coo[i][2]==0 and coo[i][3]==0)
{
sortie«"n"«endl ;
sortie«posx+80+2*coo[i][0] «" "« posy «" m"«endl ;
sortie«posx+80 «" "« posy+2*coo[i][1] «" l"«endl ;
sortie«"s"«endl ;
}

if(coo[i][2]==1 and coo[i][3]==0)
{
sortie«"n"«endl ;
sortie«posx-80+80 «" "«posy+2*coo[i][0] «" m"«endl ;
sortie«posx +80 «" "«posy+2*coo[i][1] «" l"«endl ;
sortie«"s"«endl ;
}

if(coo[i][2]==0 and coo[i][3]==1)
{
sortie«"n"«endl ;
sortie«posx+80+2*coo[i][0] «" "«posy «" m"«endl ;
sortie«posx+80+2*coo[i][1] «" "«posy+80 «" l"«endl ;
sortie«"s"«endl ;
}

if(coo[i][2]==1 and coo[i][3]==1)
{
sortie«"n"«endl ;
sortie«posx-80+80 «" "«posy+2*coo[i][0] «" m"«endl ;
sortie«posx+80+2*coo[i][1] «" "«posy+80 «" l"«endl ;
sortie«"s"«endl ;
}
}

}


//3eme cas : si l'angle est nul :
//Dans ce cas, la droite est horizontale est donc il n'y pas de problème. Une boucle classique suffit.

for (int j=1 ;j<40 ;j++){
sortie«"n"«endl ;
```

```
sortie«posx+0«" "«posy+2*j«" m"«endl;
sortie«posx+80«" "«posy+2*j«" l"«endl;
sortie«"s"«endl;
cmp++;
}
```

F Page comportant les grilles générées

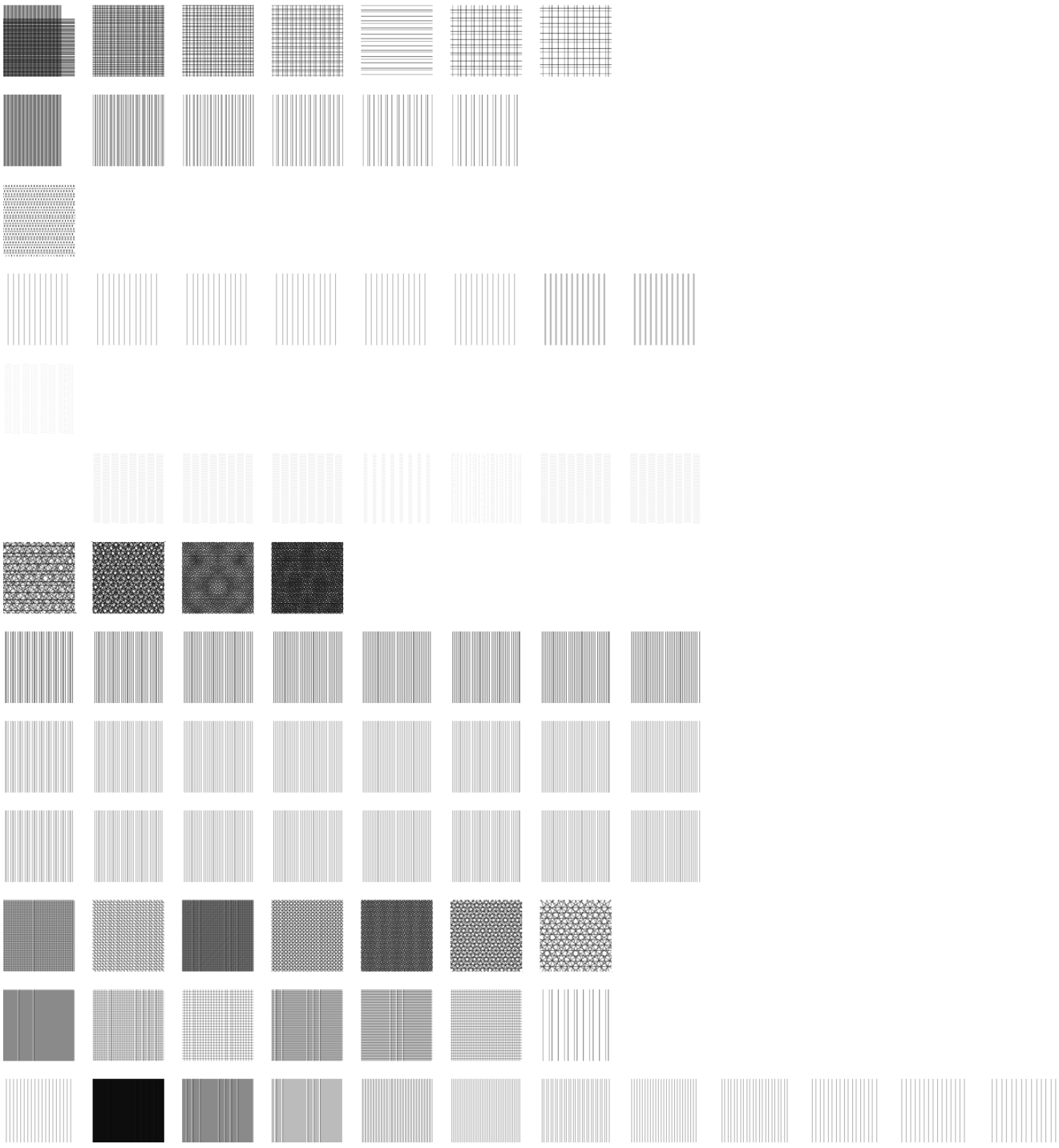


FIG. 36 – **Page comportant les grilles générées réalisées grâce, entre autre, aux méthodes expliquées dans la section 2. La faible qualité de cette figure vient du fait qu'il a fallut convertir et réduire l'image afin de pouvoir l'insérer dans ce document en latex. Comme le programme qui génère cette page fait plus de 3 500 lignes, il n'était pas possible de joindre ce programme en annexes.**