TEXTE – SOUTENANCE PROJET INFORMATIQUE – DIFFRACTION

Légende : Elève 1 Elève 2

Diapositive n°1. [Présentation] – Bonjour, je m'appelle et je suis étudiante en PTSI. Bonjour je m'appelle et je suis également étudiant en PTSI. Nous allons vous présenter le projet sur lequel nous avons travaillé, à savoir la diffraction de la lumière.

Diapositive n°2. Tout d'abord nous allons vous exposer rapidement le choix de notre projet. Ensuite nous enchainerons sur les objectifs de ce projet. Nous évoquerons également la répartition des tâches et la réalisation du projet. Nous finir cette présentation sur un bilan et les perspectives possibles pour ce dernier.

Diapositive n°3. Nous avions fait les vœux suivants : [énoncer vœux 1, vœux 2 et vœux 3]. Nos professeurs ont décidé de nous attribuer le deuxième : celui de diffraction de la lumière. Nous étions intéressés par ce projet car nous étions surpris de constater que derrière le phénomène de diffraction de la lumière existe une formule, mais aussi par la modélisation informatique de ce dernier.

Diapositive n°4. Le principal objectif de ce projet était le suivant : réussir à modéliser le phénomène de diffraction de la lumière, c'est-à-dire programmer avec Python cette double intégrale.

Diapositive n°5. [Détail de la répartition des tâches…] [Tâches 1 et 2] [Tâches 3 à 6].

Diapositive n°6. [Réalisation du projet]

a. Réalisation des fentes : Nous avons réalisé les fentes suivantes [Cliquer sur l'hyperlien] par le biais de : équation cartésienne d'un cercle, division par deux pour être au « milieu » de la matrice etc. [Expliquer brièvement pourquoi nous avons créé de minuscule fentes (pour Numpy)]

Bases	Numpy
$Taille\ matrice = 50$	$Taille\ matrice = 2000$
$Côté\ du\ carré = 8$	$Côté\ du\ carré = 8$
$Rayon\ du\ cercle = 5$	$Rayon\ du\ cercle = 5$
$Longueur\ du\ rectangle = 10$	$Longueur\ du\ rectangle = 5$
$Largeur\ du\ rectangle = 4$	$Largeur\ du\ rectangle = 3$
$Ecart\ entre\ les\ rectangle = 6$	$Ecart\ entre\ les\ rectangle = 30$
$Côté\ des\ petits\ carrés = 5$	$Côté\ des\ petits\ carrés = 5$
$Ecart\ entre\ les\ carrés = 10$	$Ecart\ entre\ les\ carrés = 35$

b. Pour programmer la double intégrale [Cliquer sur l'hyperlien pour une meilleure compréhension], nous l'avons « transformée » en somme qui parcourt et la fente de diffraction et l'écran sur lequel vient la figure de diffraction, qui ne sont rien d'autre que des matrices en soit.

c. Nous avons ensuite testé cette fonction et obtenu les résultats suivants [Cliquer sur l'hyperlien ; commenter ne pas oublier la deuxième diapositive].

d. J'ai également réalisé des recherches sur la fonction Fast Fourrier Transform de Numpy, couplé à la fonction fftshift qui permet de recentrer la figure de diffraction de la lumière, car sinon elle se retrouve dans les quatre coins de l'écran. Ce couple de fonctions effectue également des figures de diffraction de la même manière mais en des temps records.
e. En effet, nous avons créé un module qui nous a permis de comparer précisément les temps de calcul des deux méthodes [Cliquer sur l'hyperlien ; faire défiler les diapositives] et le résultat est sans appel : Numpy est 60'000 fois plus rapide pour le calcul d'une figure de diffraction avec une matrice de taille 100.
f. Elève a également créé une batterie d'images pour créer des GIFs animés. En effet, prenons l'exemple des deux fentes rectangulaires [Cliquer sur l'hyperlien de la diapositive Animation puis sur celui pour ouvrir l'animation]. J'ai pris une matrice de taille 20 car la complexité de ce programme pour cette fente et de l'ordre du $O(n^3)$. J'ai ensuite fait varier chaque paramètre de cette fente, à savoir : la longueur et la largeur de chaque rectangle, ainsi que l'écart entre les deux. Dès qu'un paramètre change, Python recalcul la figure de diffraction associée et me génère une image grâce au module matplotlib. Une fois que j'avais toutes les images dont j'avais besoin, je me suis rendu sur un site internet créant des GIFs animés à partir de plusieurs images et ce pour toutes les fentes.
g. Puis ensuite nous avons commenté les différents programmes Python.

Diapositive n°7.

a. En conclusion, nous avons appris à créer des modules Python et à les utiliser dans des scripts externes pour une meilleure lisibilité du script principal.
b. Nous avons également compris réellement l'importance de la complexité d'un programme, en effet la rapidité de Numpy face à notre programme est sans appel.
c. À cela s'ajoute l'automatisation de tâches telles que la création de répertoires avec Python, ou encore la sauvegarde d'images.
d. Nous avons découvert d'autres fonctionnalités de Numpy, en autre les fonctions fft et fftshit.
e. Mais aussi la manière de modéliser le phénomène de diffraction.
f. Néanmoins, nous pourrions améliorer le programme en ajoutant une interface graphique et un curseur pour générer des figures de diffraction de la lumière à la demande.
g. Mais pour que cela soit réalisable correctement, il faudrait alors réduire la complexité de notre programme qui est de l'ordre de $O(n^4)$ cela est dû à la création des fentes de diffraction, mais cette complexité se ressent surtout lors de l'utilisation de la fonction diffraction.
h. Également, de manière plus optionnelle, nous pourrions aussi ajouter d'autres fentes, plus fantaisie par exemple.

i. Vous trouverez tous les liens qui nous ont été utiles sur la dernière diapositive. Merci pour votre attention. Si vous avez des questions, nous vous écoutons.