Calcul numérique avec Scilab

Nous allons utiliser le logiciel de calcul SciLab, un clone open-source de MATLAB. Il peut être téléchargé ici:

<http://www.scilab.org/fr/download/6.0.1>

Pour ceux qui ne connaissent ni MATLAB ni SciLab il vaut la peine de regarder ce tutoriel [pdf] : « Scilab pour les vrais débutants »

http://www.scilab.org/fr/content/download/849/7897/file/Scilab_debutants.pdf [fr]

Pour ceux qui ont soif de plus : Une liste extensive de documentation et tutoriels pour SciLab:

<https://wiki.scilab.org/Tutorials>

Cours :

- Démarrer avec Scilab : On va feuilleter ensemble le tutoriel « Scilab pour les vrais débutants » + démonstration sur l'ordinateur du prof.
- Faire des transformées de Fourier en Scilab en utilisant les fonctions `fft`, `ifft`, `fftshift` et `ifftshift` : On regarde Ex. 1,2 , démonstrations sur l'ordinateur du prof.
- Présentation des sujets pour les TPs Scilabs.

TP Exercices Scilab obligatoires :

Ex. 1 : Transformée de Fourier Discrète avec FFT

Ex. 2 : Propriété de translation de la FFT

Ex. 3 : Relation entre FFT et TF continue

Ex. 4 : Comprendre l'oscilloscope numérique

Ex. 5 : Transformée de Fourier discrète bi-dimensionnelle avec la FFT

Ex. 6 : Simple illusion optique – superposition d'une image à basses fréquences spatiales avec une image à hautes fréquences spatiales

Ex. 7 : Optique de Fourier avec la FFT

Ex. 8 : Filtre analogique de 1^{er} ordre

Ex. 9 : Filtre analogique de 2^{ème} ordre

TP Projets Scilab indépendants / Avancés (pour les acharnés) :

1: Split-step Fourier method

https://www.researchgate.net/publication/281441538_An_introduction_to_the_Split_Step_Fourier_Method_using_MATLAB

2: Holographie digitale

<http://wavefrontshaping.net/index.php/63-community/tutorials/phase-measurement/94-off-axis-holography>

3: Façonnage super-Nyquist https://en.wikipedia.org/wiki/Pulse_shaping

4: Orthogonal frequency-division multiplexing (OFDM)

https://en.wikipedia.org/wiki/Orthogonal_frequency-division_multiplexing

5: Propagation de Fresnel https://en.wikipedia.org/wiki/Fresnel_diffraction

6: Équation de la chaleur en 2 et 3 dimensions https://en.wikipedia.org/wiki/Heat_equation

7: Sous-échantillonnage <https://en.wikipedia.org/wiki/Undersampling>

8: Algorithme de Gerchberg-Saxton https://en.wikipedia.org/wiki/Gerchberg%E2%80%93Saxton_algorithm

9: Coder une TFD et la comparer avec la FFT cf cours

10: Dynamic range compression d'une piste audio

https://en.wikipedia.org/wiki/Dynamic_range_compression

Exemple 1: Transformée de Fourier Discrète avec FFT

- Choisir le nombre d'échantillons N . Une puissance entière de 2 est approprié.
- Créer n , un vecteur contenant les numéros d'échantillon.
- Créer k , un vecteur contenant les numéros d'échantillon dans le spectre.
- Créer f_{delta} , un vecteur dont les valeurs suivent une fonction delta discret.
- Utilisant `fft`, calculer $f_{\text{delta_fft}}$, la transformé de Fourier discrète de f_{delta} .
- Utilisant `plot`, visualiser f_{delta} et $f_{\text{delta_fft}}$; vérifier que l'allure de $f_{\text{delta_fft}}$ est celle attendue.
- Utilisant `ifft`, vérifier qu'on retrouve f_{delta} en prenant la transforme inverse de $f_{\text{delta_fft}}$.

```
N = 1024;
n = -N/2:N/2-1;
k = -N/2:N/2-1;
n0 = 0;
fdelta = zeros(1,N); fdelta(find(n==n0)) = N;
fdelta_fft = fftshift(fft(fftshift(fdelta)));
figure(1);
subplot(2,1,1), plot(k, abs(fdelta_fft), 'k.-');
subplot(2,1,2), plot(k, phasemag(fdelta_fft,'m'), 'k.-');
fdelta_test = ifftshift(ifft(ifftshift(fdelta_fft)));
sum(abs(fdelta_test - fdelta))
```

Alternatif:

```
N = 1024;
n = [0:N/2-1 -N/2:-1];
k = [0:N/2-1 -N/2:-1];
n0 = 0;
fdelta = zeros(1,N); fdelta(find(n==n0)) = N;
fdelta_fft = fft(fdelta);
figure(2);
subplot(2,1,1), plot(k, abs(fdelta_fft), 'k.-');
subplot(2,1,2), plot(k, phasemag(fdelta_fft,'m'), 'k.-');
fdelta_test = ifft(fdelta_fft);
sum(abs(fdelta_test - fdelta))
```

Exemple 2: Définir les axes temporelle et fréquentielle en FFT

```
N = 1024; //Nombre d'échantillons [scalaire]
n = -N/2:N/2-1; //Numéro d'échantillon dans l'espace direct [vecteur 1xN, sans
unité]
Delta = 1e-3; //Pas d'échantillonnage [scalaire, unité s]
t = Delta*n; //Axe temporelle [vecteur 1xN, unité s]
k = -N/2:N/2-1; //Numéro d'échantillon dans l'espace inverse [vecteur 1xN, sans
unité]
nu = 1/(N*Delta)*k; //Axe fréquentielle [vecteur 1xN, unité Hz]
```