

42-Utilisation de la FFT

Le théorème de Fourier

Le mathématicien français Joseph Fourier a démontré que toute onde périodique peut être exprimée comme somme de sinusoides harmoniquement liées, chacune ayant sa propre amplitude et phase. Étant donné la représentation numérique d'une onde périodique, on peut utiliser une formule connue sous le nom de transformée de Fourier discrète (DFT) pour calculer la fréquence, la phase et l'amplitude de ses composants sinusoidaux. La DFT transforme essentiellement la représentation d'une onde sonore dans le domaine temporel en un spectre dans le domaine fréquentiel. Ce spectre peut également être retransformé en une forme d'onde dans le domaine temporel.

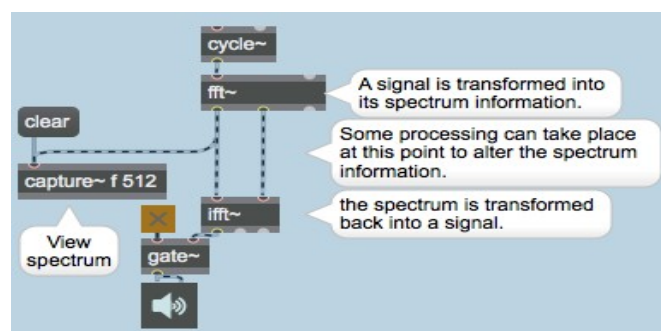
En général, la transformée de Fourier est utilisée sur une petite "tranche de temps", qui correspond idéalement à exactement un cycle de l'onde analysée. Pour effectuer cette opération sur des sons du «monde réel» - qui ne sont presque toujours pas strictement périodiques, et dont la fréquence peut être inconnue - on peut effectuer la DFT sur des tranches de temps consécutives pour avoir une idée de la façon dont le spectre évolue dans le temps.

Si le nombre d'échantillons numériques dans chaque tranche de temps (ou *trame*) est une puissance de 2, on peut utiliser une version plus rapide de la DFT appelée transformation de Fourier rapide (FFT). La formule de la FFT est encapsulée dans l'objet *fft* ~. Les mathématiques de la transformée de Fourier vont bien au-delà de la portée de ce manuel, mais ce chapitre du didacticiel montrera comment utiliser l'objet *fft* ~ pour l'analyse du signal et la resynthèse des signaux.

Spectre d'un signal: *fft* ~

fft ~ reçoit un signal dans son entrée. Pour chaque tranche de temps qu'il reçoit (512 échantillons par défaut), il envoie un signal de même longueur listant la quantité d'énergie dans chaque région de fréquence. Le signal qui sort de *fft* ~ n'est pas quelque chose que vous voudrez probablement écouter. Il s'agit plutôt d'une liste d'amplitudes relatives de 512 bandes de fréquences différentes dans le signal reçu. Cette "liste" est exactement de la même longueur que les échantillons reçus dans chaque tranche de temps, elle sort donc au même rythme que le signal entrant. Le signal qui sort de *fft* ~ est une analyse dans le domaine fréquentiel des échantillons reçus dans la tranche de temps précédente.

En fait, l'analyse produite par *fft*~ est constituée de deux signaux parallèles. C'est parce que les mathématiques impliquées produisent des nombres complexes, qui ont une composante réelle et imaginaire. Les composantes réelles sont sorties par la sortie de gauche et les composants imaginaires par celle du milieu. Les deux signaux sont nécessaires pour le traitement spectral.



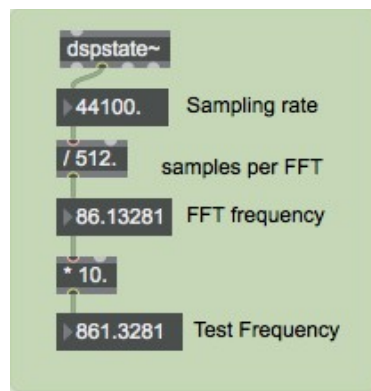
Le signal sortant de *fft* ~ est une information spectrale, pas un signal temporel

Bien que la transformée sorte de *fft* ~ sous la forme d'un **signal**, il ne s'agit pas un signal dans le domaine temporel. Le seul objet qui "comprend" ce **signal** spécial est l'objet *ifft* ~, qui effectue une FFT inverse sur le spectre et le retransforme en une forme d'onde dans le domaine temporel.

Avec l'objet *plot* ~, vous pouvez récupérer la sortie de *fft* ~ et afficher l'analyse fréquentielle d'un signal. (Le paramétrage de *plot* ~ à cet effet est abordé plus loin.) *plot* ~ est le graphique situé au bas du patch.

- Cliquez sur l'un des objets *ezdac* ~ pour activer l'audio.

Lorsque le son est activé, *dspstate* ~ envoie le taux d'échantillonnage MSP par sa sortie centrale. Nous utilisons ce nombre pour calculer une fréquence qui a une période d'exactly 512 échantillons. C'est la fréquence fondamentale de la FFT elle-même. Si nous envoyons une onde de cette fréquence dans *fft* ~, chaque tranche de temps contiendra exactement un cycle de la forme d'onde. Nous allons en fait utiliser une onde cosinusoidale à dix fois cette fréquence comme tonalité de test pour notre analyse, comme indiqué ci-dessous.



La tonalité de test est égale à 10 fois la fréquence de base de la tranche de temps FFT

Le coin supérieur gauche de la fenêtre Patcher montre une utilisation très simple de *fft* ~. L'analyse est stockée dans un objet *capture*~, et un objet *ifft* ~ transforme l'analyse en un signal audio. (Normalement, vous ne transformeriez pas et n'inverseriez pas un signal audio sans aucune raison de ce genre. Le *ifft*~ est utilisé dans ce patch simplement pour démontrer que le processus d'analyse-resynthèse fonctionne.)

- Cliquez sur le *toggle* dans la partie bleue du patch pour entendre le son resynthétisé. Cliquez à nouveau sur le *toggle* pour fermer la *gate* ~. Maintenant, regardez le tracé en bas de la *fft* ~.

plot~ montre un pic unique à 861,3 Hz. Si vous cliquez sur le message **clear** et que vous double-cliquez sur l'objet *capture*~, vous verrez certains des nombres que la *fft* ~ émet. Chacun des 512 nombres représente une harmonique de la fréquence FFT elle-même, commençant par la 0ème harmonique (0 Hz). L'analyse montre de l'énergie dans le onzième nombre, qui représente le 10ème harmonique de la FFT, **10/512** le taux d'échantillonnage - précisément notre fréquence de test. L'analyse montre également de l'énergie au 10ème nombre à partir de la fin (faites défiler pour voir), qui représente **502/512** la fréquence d'échantillonnage. Cette fréquence dépasse le taux de Nyquist et est en fait équivalente à **-10/512** de la fréquence d'échantillonnage. (Cette composante n'est pas visible dans l'affichage de *plot*~ qui est limité à la moitié de la fréquence d'échantillonnage).

Détail technique: une FFT divise l'ensemble de la gamme de fréquences disponible en autant de bandes (souvent appelées «bins») qu'il y a d'échantillons dans chaque tranche de temps. Par conséquent, chaque ensemble de 512 nombres sortant de *fft* ~ représente 512 divisions de la gamme de fréquences allant de 0 à la fréquence d'échantillonnage. Le premier nombre représente l'énergie à 0 Hz, le deuxième nombre représente l'énergie à $1/512$ de la fréquence d'échantillonnage, le troisième nombre représente l'énergie à $2/512$ de la fréquence d'échantillonnage, et ainsi de suite.

Notez qu'une fois que nous atteignons le taux de Nyquist sur le 257ème nombre ($256/512$ du taux d'échantillonnage), tous les nombres suivants sont repliés vers le bas du taux de Nyquist. Une autre façon de voir les choses est que ces nombres représentent des fréquences négatives qui sont maintenant ascendantes par rapport au taux de Nyquist (négatif). Ainsi, le 258ème nombre est l'énergie au taux de Nyquist moins $1/512$ du taux d'échantillonnage (ce qui pourrait être également considéré comme $-255/512$ du taux d'échantillonnage). Dans notre exemple, nous voyons de l'énergie dans la 11ème région de fréquence ($10/512$ le taux d'échantillonnage) et dans la 503ème région de fréquence ($-256/512 - -246/512 = -10/512$ du taux d'échantillonnage).

Il semble que *fft* ~ ait correctement analysé le signal. Il y a juste un problème ...

Problèmes pratiques de la FFT

La FFT suppose que les échantillons analysés comprennent un cycle d'une onde périodique. Dans notre exemple, l'onde cosinoïdale était la 10e harmonique de la fréquence fondamentale de la FFT, et cela a donc bien fonctionné. Cependant, dans la plupart des cas, les 512 échantillons de la FFT ne correspondent pas exactement un cycle de l'onde. Dans ce cas, la FFT analyse toujours les 512 échantillons comme s'il s'agissait d'un cycle d'une forme d'onde, et rapporte le spectre de cette onde. Une telle analyse contient de nombreuses fréquences parasites qui ne sont pas réellement présentes dans le signal.

- Fermez la fenêtre de texte de *capture* ~. Avec le son étant toujours actif, réglez la boîte de *nombre* 'Test Frequency' sur **1000**. Ceci déclenche également le message **clear** situé dans le coin supérieur gauche du patch pour vider l'objet *capture*~ de son contenu antérieur. Double-cliquez une nouvelle fois sur *capture*~ pour voir son nouveau contenu.

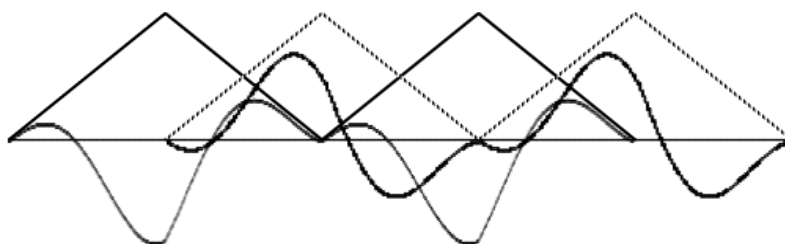
L'analyse du son à 1 000 Hz montre effectivement une plus grande énergie à 1 000 Hz - dans les 12ème et 13ème régions de fréquence si votre taux d'échantillonnage MSP est de 44 100 Hz - mais elle montre aussi de l'énergie dans pratiquement toutes les autres régions. C'est parce que la forme d'onde qu'il a analysée n'est plus une sinusoïde. (Un nombre exact de cycles ne tient pas précisément dans les 512 échantillons.) Toute autre énergie montrée dans cette FFT est un artefact de l'interprétation «incorrecte» de ces 512 échantillons comme une période de la forme d'onde correcte. De plus, si vous fermez l'objet *capture* ~, que vous appuyez sur *clear* et que vous le rouvrez, vous verrez que les nombres ont tous changé, car la relation de phase entre 1000Hz et 947.5 (la 11ème harmonique de la fréquence de la *fft*) change constamment.

Tout cela sera visible dans *plot*~, qui présente un large pic dont les jupes changent constamment.

Pour résoudre ce problème, nous pouvons essayer d'effiler les extrémités de chaque tranche de temps en lui appliquant une enveloppe d'amplitude, et utiliser des tranches de temps qui se chevauchent pour compenser l'utilisation de l'enveloppe.

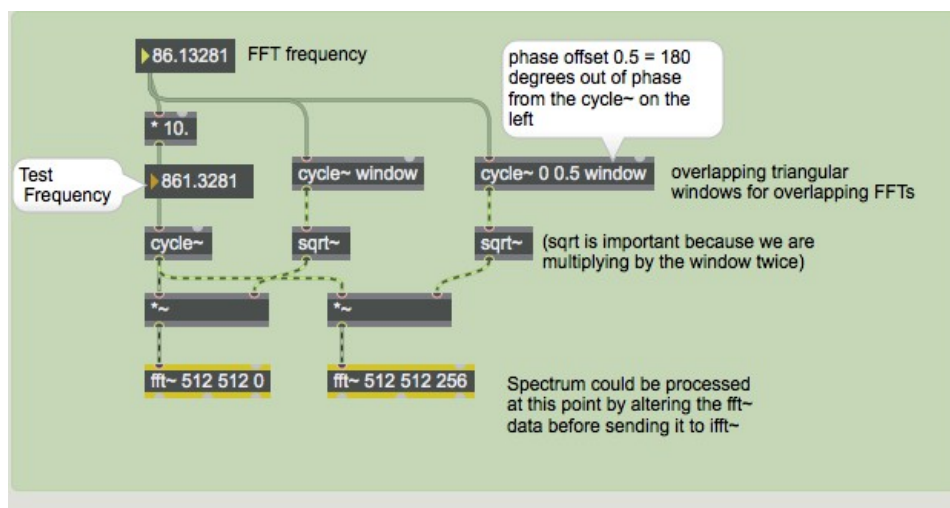
Overlapping FFTs (superposées)

La partie droite du patch du didacticiel adopte cette approche d'utilisation des tranches de temps superposées, et applique une enveloppe d'amplitude triangulaire à chaque tranche *avant* de l'analyser, puis *après* l'avoir resynthétisée. (D'autres formes d'enveloppe d'amplitude sont souvent utilisées pour ce processus, mais la fenêtre triangulaire est simple et assez efficace.) De cette façon, l'objet *fft* ~ visualise chaque tranche de temps à travers une fenêtre triangulaire dont les extrémités se rétrécissent, filtrant ainsi un grand nombre de fausses fréquences qui seraient introduites par des discontinuités. Cette technique est connue sous le nom de *fenêtrage*.



Fenêtres triangulaires se chevauchant (enveloppes) appliquées à une onde cosinus de 100 Hz

Pour réaliser ce fenêtrage et ce chevauchement des tranches de temps, nous devons effectuer deux FFT, dont l'une est décalée de 256 échantillons plus tard que l'autre. (Notez que cette partie du patch ne fonctionnera que si la taille actuelle de votre vecteur de signal MSP est de 256 ou moins, puisque la *fft* ~ ne peut être décalée que par un multiple de la taille du vecteur.) L'offset d'une FFT peut être donné sous forme de (troisième) argument introduit à *fft* ~, comme cela est fait pour l'objet *fft* ~ à droite. Il en résulte des tranches de temps qui se chevauchent.



Une FFT est prise 256 échantillons plus tard que l'autre

Le fenêtrage est réalisé en multipliant le signal par une forme d'onde triangulaire (stockée dans l'objet *buffer* ~ - double-cliquez pour afficher son contenu) qui revient à la même fréquence que la FFT - une fois tous les 512 échantillons. La fenêtre est décalée d'un 1/2 cycle (256 échantillons) pour la deuxième *fft* ~. Remarquez également que, comme nous appliquerons l'enveloppe d'amplitude deux fois (une fois avant la *fft* ~ et une fois encore après l'*ifft* ~), nous prenons la racine carrée des valeurs d'enveloppe, afin d'éviter toute modulation d'amplitude indésirable résultant de nos enveloppes (nous voulons que les enveloppes qui se chevauchent se croisent uniformément et que leur somme soit toujours égale à 1).

- Cliquez sur le *toggle* intitulé «Afficher l'analyse en fenêtre». Ceci ajoutera la fft fenêtrée (en rouge) à l'affichage.

Comme avec la FFT non fenêtrée, l'énergie est maximale autour de 1000 Hz, mais ici l'énergie (parasite) dans toutes les autres régions de fréquence est considérablement réduite par rapport à la version non fenêtrée.

Affichage des spectres avec *plot* ~

L'objet *plot* ~ est conçu pour afficher des graphiques de données listées ou de signaux audio. Avec ces derniers, il fonctionne un peu comme *scope* ~, où la portion de forme d'onde affichée est déterminée par le nombre de points dans l'affichage. Dans *plot* ~, ce nombre est défini par un message **numpoints**. *plot* ~ dispose de nombreuses options, en particulier si vous souhaitez ajouter des lignes de grille et des titres à l'affichage. Heureusement, il existe quelques prototypes disponibles pour *plot* ~, accessibles via le bouton situé sur le bord gauche de l'écran. Les prototypes comprennent plusieurs affichages spectraux complets avec des lignes de grille et des étiquettes.

Il y a un peu d'assemblage nécessaire avant que *plot* ~ puisse afficher le spectre fft. Ouvrez le sub-patch `Display_analysis` pour voir ce qui est nécessaire.

- Les signaux de *fft* ~ peuvent atteindre des valeurs très élevées (aussi élevées que la taille de l'image) et doivent être réduits à un maximum de 1,0 pour *plot* ~. D'où la division par 512.
- La **puissance** dans chaque bin est représentée par un nombre complexe. La magnitude d'un nombre complexe est la racine carrée de la somme des carrés des parties réelles et imaginaires, donc chaque signal de la *fft* ~ est élevé au carré avant d'être additionné. Si nous omettons la racine carrée, nous obtenons une échelle exponentielle, qu'il convient de représenter en dB.
- *Plot* ~ n'a pas de mécanisme de déclenchement comme le font *scope* ~ et *spectroscope* ~, donc le signal de fft doit être stabilisé avant de pouvoir être affiché. L'objet *vecbral* ~ est idéal pour cela, car il aligne les noms entrants avec les indices dans les entrées de gauche. La sortie droite de *fft* ~ produit une rampe de nombres de bin allant de 0 à la taille de la fenêtre, parfaite pour *vecbral* ~. *vecbral* ~ peut également faire la moyenne des valeurs sur plusieurs images, ce qui rend l'affichage plus lisse.

La plupart du temps, nous utiliserons *spectroscope* ~ comme affichage spectral, mais si vous souhaitez des fonctionnalités personnalisées ou une résolution très élevée, *plot* ~ et *fft* ~ vous permettent de créer le vôtre.

Traitement du signal à l'aide de la FFT

Dans ce patch, nous avons utilisé l'objet *fft* ~ pour visualiser et analyser un signal, et pour démontrer l'efficacité du fenêtrage du signal et de l'utilisation de FFT superposés. Cependant, on pourrait aussi écrire un patch qui modifie les valeurs du signal sortant de *fft* ~, puis envoie l'analyse modifiée à *ifft* ~ pour la resynthèse. Ce type de traitement utilisant des tranches de temps superposées et fenêtrées est appelé transformée de Fourier à court terme (STFT) et constitue la base du traitement audio dans le domaine de la fréquence. Nous aborderons une façon plus simple d'utiliser STFT dans le didacticiel 26.

Le fenêtrage, en plus d'être important pour réduire les fausses fréquences provenant des discontinuités dans la forme d'onde d'entrée, comme nous l'avons déjà vu, est également important pour lisser les discontinuités qui se produisent dans la forme d'onde resynthétisée dans le domaine

temporel provenant de *ifft* ~. C'est pourquoi nous devons fenêtrer le signal temporel à la fois en entrée et en sortie. Dans ce tutoriel, nous n'obtiendrions de telles discontinuités en sortie que si nous modifions le signal entre les objets *fft* ~ et *ifft* ~.

Résumé

La transformée de Fourier rapide (FFT) est un algorithme permettant de transformer un signal numérique dans le domaine temporel en une représentation dans le domaine fréquentiel de l'amplitude relative des différentes zones de fréquence du signal. Une FFT est calculée en utilisant un extrait relativement petit d'un signal, généralement une tranche de temps de 512 ou 1024 échantillons. Pour analyser un signal plus long, on effectue plusieurs FFT en utilisant des tranches de temps consécutives (ou qui se chevauchent).

L'objet *fft* ~ effectue une FFT sur le signal qu'il reçoit et envoie (également sous la forme d'un signal) une analyse dans le domaine fréquentiel du signal reçu. Le seul objet qui comprend la sortie de *fft* ~ est *ifft* ~ qui exécute une FFT inverse pour synthétiser un signal dans le domaine temporel basé sur l'information dans le domaine fréquentiel. On pourrait modifier le signal lors de son passage de *fft* ~ à *ifft* ~, afin de changer le spectre.

La FFT ne fonctionne parfaitement que lorsqu'elle analyse exactement un cycle (ou un nombre entier de cycles) d'un son. Pour réduire les artefacts produits lorsque ce n'est pas le cas, on peut fenêtrer le signal analysé en appliquant une enveloppe d'amplitude pour atténuer les extrémités de chaque tranche de temps. L'enveloppe d'amplitude peut être appliquée en multipliant le signal par l'utilisation d'un objet *cycle* ~ pour lire une fonction de fenêtrage à partir d'un *buffer* ~ de manière répétée au même rythme que la FFT elle-même (c'est-à-dire une fois par tranche de temps). Pour éliminer tout artefact résultant de la modification du signal dans le domaine fréquentiel, nous devons également appliquer la même enveloppe à la sortie de l'*ifft* ~.