

19-synthèse - mise en forme d'onde

Dans ce didacticiel, nous allons étudier un attribut latent (mais très utile) des échantillons, à savoir qu'ils peuvent être utilisés comme tables de consultation pour transformer la forme d'autres formes d'onde. Ce processus s'appelle la mise en forme d'onde et est utilisé en synthèse pour générer des spectres complexes à partir d'une entrée sinusoïdale. C'est également la technique du traitement du signal de base qui se cache derrière de nombreux types de distorsion dépendant de l'amplitude et qui peut être utilisée pour modéliser les non-linéarités de différents types d'amplificateurs.

Utilisation d'une table d'onde stockée comme fonction de transfert

Jetez un coup d'œil au patcheur du tutoriel. Le circuit de base de génération du son en haut à gauche du patcheur devrait vous sembler familier, avec un nouvel objet (*lookup* ~) inséré dans la chaîne. Nous avons un objet *cycle* ~ passant par un amplificateur (* ~) jusqu'à l'*ezdac* ~.

- Activez l'objet *ezdac* ~ et ajustez la boîte de *nombre* intitulée amplitude . Vous devriez entendre une onde sinusoïdale à 220 Hz.

Le nouvel objet de cette chaîne de signal semble d'abord ne rien faire et, d'après ce que nous entendons, ce n'est pas ... pour le moment. L'objet *lookup* ~ interprète un morceau de mémoire d'échantillon stocké dans un objet *buffer* ~ comme une fonction de transfert, le début de l'échantillon utilisé représentant les valeurs d'entrée de **-1** et la fin de l'échantillon utilisé représentant les valeurs de **1**. Les valeurs entrantes sont mises à l'échelle. sur cet axe X, et l'audio résultant provient des valeurs correspondantes sur l'axe Y.

Dans notre patch, le *buffer* ~ appelé **waveform** sert de **table de consultation** (ou fonction de transfert) pour l'onde sinusoïdale entrante. Lorsque l'objet *cycle* ~ génère un **-1**, par exemple, **quelle que soit la valeur d'échantillon** se trouvant au début du *buffer* ~ sort de l'objet *lookup* ~. Lorsque notre *cycle* ~ atteint **1**, l'objet *lookup* ~ lit depuis la fin du *buffer* ~ pour trouver son échantillon sortant.

- Double-cliquez sur l'objet *buffer* ~ en bas du patcheur du didacticiel. Remarquez que la forme d'onde chargée est une simple rampe. Comme notre forme d'onde, l'échantillon dans le *buffer*~, est une rampe linéaire avec le début de l'échantillon à **-1** et la fin à **1**, notre objet *cycle*~ sonne inchangé.

L'objet *lookup* ~ prend trois arguments possibles: le premier est le nom du *buffer* ~ à utiliser comme forme d'onde; les deuxième et troisième sont les points de départ et de fin (en **échantillons** à utiliser dans le *buffer* ~ comme limites de la fonction de transfert. Comme nous voulons que notre *buffer* ~ ait une longueur exacte de **512** échantillons dans notre patcheur, nous l'avons créé avec l'argument de **10,66667** millisecondes. Comment avons-nous obtenu ce nombre ? À droite du patch, vous pouvez voir l'objet *sampstoms* ~, ce qui nous permet de convertir des échantillons en millisecondes.

L'objet *Waveform*~

Examinez l'objet graphique en haut du patcheur du didacticiel. Notez qu'il contient la même forme que celle chargée dans notre *buffer* ~. L'objet *waveform* ~ nous permet de visualiser, de sélectionner des régions et de modifier directement le contenu d'un *buffer* ~ avec un outil de dessin.

- À l'aide de votre souris, gribouillez dans l'affichage de l'objet *waveform* ~. Remarquez comment les différentes formes affectent le son produit. Essayez de dessiner des courbes lisses, puis des courbes irrégulières, puis des courbes avec beaucoup de plateaux (lignes horizontales droites).

Remarquez que même de légères variations dans la forme ont un impact considérable sur le spectre généré par l'objet *lookup* ~.

L'objet *waveform* ~ fonctionne en mode dessin, où vous pouvez littéralement modifier un échantillon chargé dans un *buffer* ~ avec votre souris. D'autres modes vous permettent de sélectionner des régions, dont les limites peuvent être utilisées comme messages Max pour d'autres objets.

Réglage des valeurs d'échantillon avec Max messages: *peek* ~

- Cliquez sur le bouton marqué **A** du patcheur du didacticiel. Notre forme d'onde (et notre son) devrait revenir à la normale.

La façon dont nous avons obtenu la forme d'onde par défaut dans notre patcheur de didacticiel est à travers la logique contrôlée par l'*uzi* sous le *button* étiqueté **A**. L'objet *uzi*, vous vous en souvenez peut-être, génère beaucoup de données instantanément en fonction de son argument. La sortie droite de l'objet génère une rampe numérique allant de **1** à son argument. Nous avons soustrait **1** de cette valeur pour générer un flux de nombres Max compris entre **0** et **511** lorsque vous cliquez sur le *button*. Ces nombres sont ensuite envoyés aux entrées centrales et gauches d'un objet *peek* ~. L'entrée centrale reçoit un nombre qui a été mis à l'échelle dans la plage de **-1** à **1** (via l'objet *scale*), puis l'entrée gauche reçoit le nombre inchangé. L'objet *peek* ~ nous permet de définir manuellement les valeurs de l'échantillon dans un *buffer* ~ via des messages Max. L'entrée gauche (qui est "chaude" et qui effectue réellement l'opération) définit l'échantillon que nous changeons à la valeur la plus récente reçue par l'entrée centrale. Dans la section **A**, l'objet *uzi* génère un flux de nombres qui remplit tout notre *buffer* ~ avec une rampe ascendante, par ex.

Valeur d'échantillon

0 -1.

1 -0.996086

2 -0,992172

3 -0,988258

4 -0,984344

5 -0,980431

...

508 0,988258

509 0,992172

510 0,996086

511 1.

Pour les geeks de mathématiques dans la salle

- Cliquez sur les autres objets *button* du patcheur (étiquetés **B**, **C** et **D**). Remarquez comment la forme d'onde dans le *buffer* ~ change, et comment cela affecte le son. Chaque forme d'onde semble multiplier la fréquence du son généré par l'objet *cycle* ~. La logique en **B** fait sonner notre onde à **220 Hz** à **440**, **C** à **660** et **D** à **880**.

Les équations que les objets *expr* effectuent dans ces parties du patch génèrent des types spéciaux de fonctions de transfert appelées *polynômes de Chebyshev*. Ces fonctions sont intéressantes car elles ont la capacité de transformer une entrée sinusoïdale en différents multiples harmoniques. Les quatre polynômes de Chebyshev dans notre didacticiel sont les suivants:

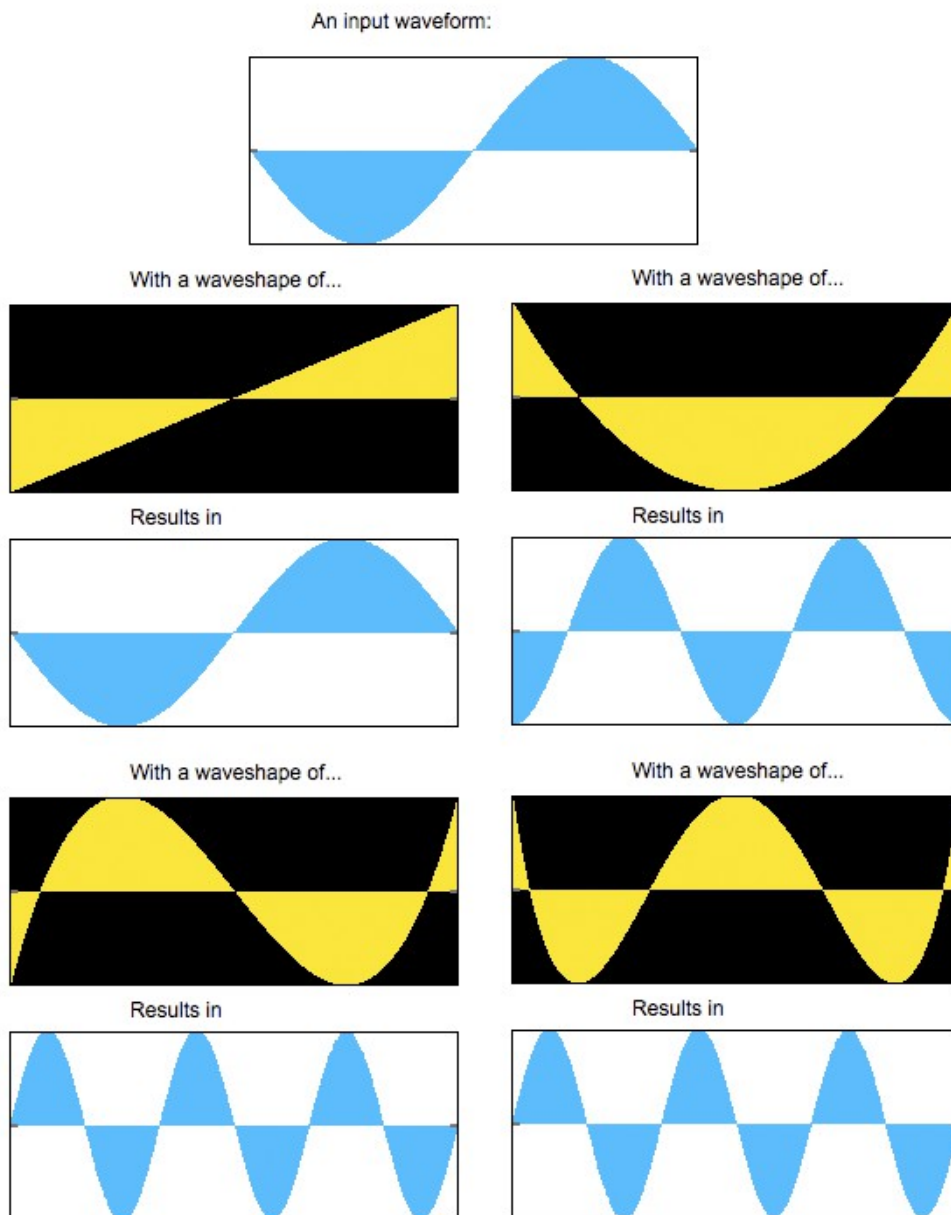
$y = x$ (l'objet *uzi A*, laisse l'entrée inchangée)

$y = 2 * x^2 - 1$ (objet *uzi B*, double la fréquence)

$y = 4 * x^3 - 3 * x$ (l'objet *uzi C* triple la fréquence)

$y = 8 * x^4 - 8 * x^2 + 1$ (objet *uzi D*, quadruple la fréquence)

En pratique, cela ressemble à ceci:



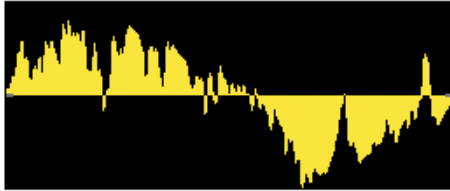
Nos quatre polynômes de Chebyshev et leur effet sur une entrée de cosinus

- Cliquez sur les messages **read** dans la zone **E** du patcheur. Ceux-ci vont charger 512 fichiers audio dans notre *buffer* ~ de formes d'onde. Notez leur effet sur le son:

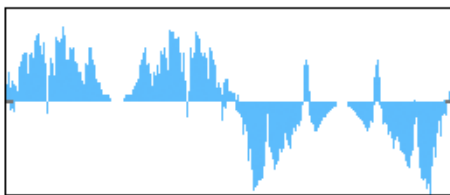
An input waveform:



With a waveshape of...



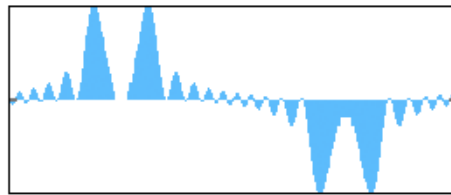
Results in



With a waveshape of...



Results in



Mise en forme d'onde via gtr512.aiff et blp512.aiff, respectivement.

- Maintenant que vous en savez un peu plus sur le comportement attendu de la distorsion de mise en forme d'onde, dessinez dans l'objet *waveform~* pour voir quel genre de résultats vous pouvez obtenir.

Résumé

L'objet *lookup~* vous permet d'utiliser un *buffer~* comme fonction de transfert pour exécuter un processus appelé **waveshaping** sur un son d'entrée. Différentes formes provoquent différentes distorsions des spectres et peuvent créer des timbres très complexes. L'objet *waveform~* vous permet de visualiser et de modifier directement le contenu d'un *buffer~* MSP à l'aide de votre souris, tandis que l'objet *peek~* vous permet de définir des valeurs d'échantillon de manière programmatique avec des messages Max. Certaines fonctions de transfert (telles que les polynômes de Chebyshev) ont des propriétés particulières lorsqu'elles sont utilisées comme fonction de mise en forme d'onde sur une entrée sinusoïdale.