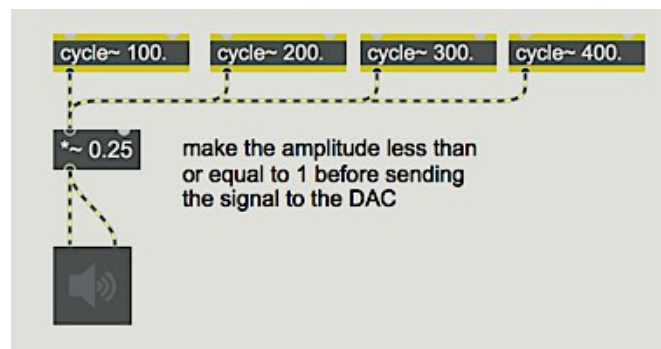


10-Synthèse additive

Dans les exemples de didacticiels proposés jusqu'à présent, nous avons synthétisé le son à l'aide de formes d'onde de base. Dans les chapitres suivants, nous allons explorer quelques autres techniques de synthèse bien connues utilisant des ondes sinusoïdales. La plupart de ces techniques sont dérivées de méthodes de synthèse analogique antérieures à l'informatique, mais elles sont néanmoins instructives et utiles.

Combinaison de sons

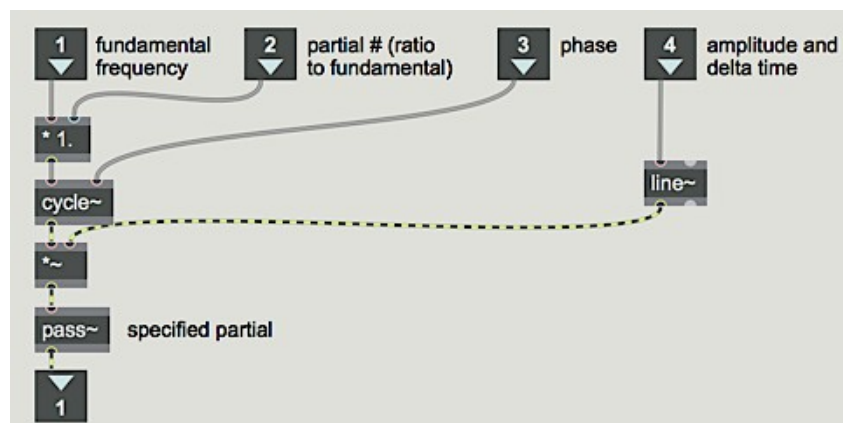
Une onde sinusoïdale contient de l'énergie à une fréquence unique. Étant donné que les sons complexes, par définition, sont composés d'énergie à plusieurs (ou à de nombreuses) fréquences différentes, une façon évidente de synthétiser des sons complexes est d'utiliser plusieurs oscillateurs sinusoïdaux et à les additionner.



Quatre sinusoides réunies pour former un son complexe

Bien sûr, vous pouvez ajouter n'importe quelle forme d'onde pour produire un son composite, mais nous nous en tiendrons aux ondes sinusoïdales dans cet exemple. La synthèse de sons complexes par l'addition d'ondes sinusoïdales est une méthode quelque peu fastidieuse, mais elle permet un contrôle complet de l'amplitude et de la fréquence de chaque composante (*partielle*) du son complexe.

Dans le patch du didacticiel, nous assemblons six oscillateurs cosinus (objets *cycle ~*), avec un contrôle indépendant sur la fréquence, l'amplitude et la phase de chacun d'entre eux. Afin de simplifier le patch, nous avons conçu une abstraction appelée **partial ~** qui permet de spécifier la fréquence de chaque partiel comme un rapport relatif à une fréquence fondamentale.



Le contenu de l'abstraction *partial ~*

Par exemple, si nous voulons qu'un partiel ait une fréquence double de celle du fondamental, il suffit de taper 2,0 comme argument (ou de l'envoyer dans la deuxième entrée). De cette façon, si plusieurs abstractions **partial** ~ reçoivent leur valeur de fréquence fondamentale (dans l'entrée gauche) de la même source, leurs fréquences relatives resteront les mêmes, même si la valeur de la fréquence fondamentale change.

Bien sûr, pour que le son soit très intéressant, les amplitudes des partiels doivent évoluer avec une relative indépendance. C'est pourquoi, dans le patch principal, nous contrôlons l'amplitude de chaque partiel avec son propre *générateur d'enveloppe*.

Générateur d'enveloppe: *function*

Dans le didacticiel de base 3, vous avez vu comment créer une enveloppe d'amplitude en envoyant une liste de paires de chiffres à un objet *line ~*, lui donnant ainsi une succession de valeurs cibles et de temps de transition. Cette idée de créer une fonction de contrôle à partir d'une série de segments de ligne est utile dans de nombreux contextes - la génération d'enveloppes d'amplitudes en est une utilisation particulièrement courante - et elle est également expliquée dans l'analyse du *didacticiel de base 5*.

L'objet *function* est d'une grande aide pour la génération de tels segments, car il vous permet de dessiner la forme souhaitée, ainsi que de définir le domaine et l'étendue de la fonction (la valeur numérique de ses dimensions sur les axes x et y). Vous pouvez dessiner une fonction simplement en cliquant avec la souris à l'endroit où vous voulez que chaque point d'arrêt apparaisse. Lorsque la *function* reçoit un **bang**, elle envoie une liste de paires valeur-temps pour sa deuxième sortie. Cette liste, lorsqu'elle est utilisée comme entrée dans l'objet *line ~*, produit un signal changeant qui correspond à la forme dessinée.

D'ailleurs, *function* est également utile à des usages autres que le signal dans Max. Elle peut être utilisée comme une table de consultation interpolante. Lorsqu'elle reçoit un nombre dans son entrée, elle considère ce nombre comme une valeur x et recherche la valeur y correspondante dans la fonction dessinée (en interpolant entre les points de séparation si nécessaire) et l'envoie par la sortie gauche.

Une variété de sons complexes

Même avec seulement six partiels, on peut créer une variété de timbres allant de sons «réalistes» ressemblant à des instruments à des combinaisons de fréquences clairement artificielles. Les réglages de quelques sons différents ont été stockés dans un objet *preset* dans le patcheur de didacticiel pour que vous puissiez les essayer. Une brève explication de chaque sonorité est fournie ci-dessous.

- Cliquez sur l'icône du haut-parleur *ezdac ~* pour activer l'audio. Cliquez sur *button* pour jouer un son. Cliquez sur l'un des préréglages stockés dans l'objet *preset* pour modifier les paramètres, puis cliquez à nouveau sur le bouton pour entendre le nouveau son.

Preset 1. Ce son n'est pas vraiment destiné à émuler un véritable instrument. Il s'agit simplement d'un ensemble de partiels harmoniquement liés, chacun d'eux ayant une enveloppe d'amplitude différente. Remarquez comment le timbre de la tonalité change légèrement au cours de sa durée, au fur et à mesure que les différents partiels passent au premier plan. Si vous ne remarquez pas vraiment ce changement de timbre, essayez de modifier la durée de la note en utilisant la boîte de *nombre* en haut du patcheur. En modifiant sa valeur à **8 000** millisecondes, par exemple, vous entendrez la note évoluer plus lentement.

Preset 2. Cette sonorité ressemble plutôt à un orgue d'église. Les partiels sont tous des octaves de la fondamentale, l'attaque est modérément rapide mais pas percutante, et l'amplitude du son ne diminue pas beaucoup au cours de la note. Vous pouvez voir et entendre que les partiels supérieurs s'éteignent plus rapidement que les partiels inférieurs.

Preset 3. Ce son est constitué de partiels harmoniques légèrement désaccordés. L'attaque est immédiate et l'amplitude décroît assez rapidement après l'attaque initiale, ce qui donne à la note un effet de percussion ou de pincement.

Preset 4. Les enveloppes d'amplitude des partiels de ce son sont dérivées de l'analyse d'une note de trompette dans le registre grave. Bien sûr, il ne s'agit que de six des nombreux partiels présents dans un véritable son de trompette.

Preset 5. Les enveloppes d'amplitude des partiels de ce son sont dérivées de la même analyse de trompette. Cependant, dans ce cas, seuls les harmoniques impaires sont utilisées. Cela crée un son plus proche de celui d'une clarinette, car la perce cylindrique d'une clarinette fait résonner les harmoniques impaires. En outre, la durée plus longue de cette note ralentit l'ensemble de l'enveloppe, ce qui lui donne l'attaque plus caractéristique de la clarinette.

Preset 6. C'est un son complètement artificiel. Le partiel le plus bas entre en premier, suivi par le sixième partiel un demi-ton plus haut. Finalement, les partiels restants entrent, avec des fréquences situées entre le premier et le sixième partiel, créant un cluster microtonal. L'effet de battement est dû à l'interférence entre ces ondes de fréquence légèrement différente.

Preset 7. Dans ce cas, les partiels sont espacés d'une seconde majeure et l'amplitude de chaque partiel monte et descend de manière à créer un effet composite de cluster de tons entiers arpégés. Bien qu'il s'agisse clairement d'un accord par tons entiers plutôt que d'un seul ton, les attaques et les diminutions progressives et superposées font que les sons fusionnent assez bien.

Preset 8. Dans ce son, les partiels suggèrent un spectre harmonique suffisamment fort pour que nous ayons encore le sentiment d'une hauteur fondamentale, mais ils sont suffisamment mal accordés pour ressembler au spectre inharmonique d'une cloche. L'attaque percussive, la décroissance rapide et les partiels variant indépendamment pendant la partie de maintien de la note sont également caractéristiques d'une cloche en métal frappée.

Notez que lorsque vous additionnez plusieurs signaux comme ceci, leur somme dépassera souvent les limites d'amplitude de l'objet *dac* ~, de sorte que l'amplitude globale doit être mise à l'échelle de manière appropriée avec un objet *~.

Expérimentation avec des sons complexes

- En utilisant ces sons comme points de départ, essayez de concevoir vos propres sons avec ce patch de synthèse additive. Variez les sonorités en modifiant la fréquence fondamentale, les partiels et la durée des sonorités prédéfinies. Vous pouvez également modifier les enveloppes en cliquant et en faisant glisser les points d'arrêt dans les objets *function*:
- Cliquez dans l'objet *function* pour créer un nouveau point d'arrêt. Si vous cliquez et faites glisser, les coordonnées x et y du point s'affichent dans la partie supérieure de l'objet et vous pouvez immédiatement déplacer le point d'arrêt à la position souhaitée.
- De même, vous pouvez cliquer et faire glisser un point d'arrêt existant pour le déplacer.
- Shift-cliquez sur un point existant pour le supprimer.

Bien que cela ne soit pas démontré dans ce tutoriel, il est également possible de créer, déplacer et supprimer des points d'arrêt dans *function* simplement en utilisant les messages Max. Voir la page de référence de l'objet *function* pour plus de détails.

Le message **setdomain**, suivi d'un nombre, modifie l'échelle de l'axe x dans la fonction sans modifier la forme de l'enveloppe. Lorsque vous modifiez le nombre dans la boîte de *nombre* 'Durée', un message **setdomain** est envoyé à *function*.

Résumé

La **synthèse additive** est le processus qui consiste à synthétiser de nouvelles sonorités complexes en ajoutant des sons simples (généralement des ondes sinusoïdales). Comme les sinusoïdes pures n'ont de l'énergie qu'à une seule fréquence, elles sont les éléments fondamentaux de la synthèse additive, mais bien sûr, n'importe quel signal peut être additionné. Le signal de la somme peut avoir besoin d'être mis à l'échelle par une valeur de signal constante inférieure à 1 afin d'empêcher qu'il ne soit par le *dac* ~.

Pour que le timbre d'un son complexe reste le même lorsque sa hauteur change, chaque partiel doit conserver sa relation avec la fréquence fondamentale. Le fait d'exprimer la fréquence de chaque partiel en termes de rapport à la fréquence fondamentale (c'est-à-dire en multipliant cette dernière) permet de conserver le spectre du son, même lorsque la fréquence fondamentale change.

Pour qu'un son complexe ait un timbre intéressant, l'amplitude des partiels doit varier avec un certain degré d'indépendance. L'objet *function* vous permet de dessiner des formes de contrôle telles que des enveloppes d'amplitude. Lorsque *function* reçoit un **bang**, il décrit cette forme à un objet *line* ~ pour générer un signal de contrôle correspondant.