

RAPPORT MATHÉMATIQUES : PROJET MUSIQUE / MATLAB

Musique : « Game of Throne, Main Theme » de Ramin Djawadi

Nous avons choisi un arrangement au piano de cette musique propose par Patrick Pietschmann disponible à l'adresse suivante :

- [Partition](#) (la synthèse a été effectuée sur les 23 premières mesures)

Pour pouvoir synthétiser cette musique nous avons dû utiliser le logiciel Matlab et créé chaque note comme décrit ci-après, puis nous les avons mis bout à bout dans une matrice (1x1 pour le mono et 2x1 pour le stéréo), quand plusieurs notes sont jouées en même temps il faut sommer leur vecteur, ce qui nécessite qu'elles aient la même durée. Enfin nous avons utilisé une fonction de Matlab, *soundsc(matrice, fréquence_d_échantillonnage)* pour pouvoir jouer la musique.

1) Extraction des données de la note (« n.m ») :

Cette fonction permet juste de facilité l'écriture du programme, elle est appelée pour la création d'une note en lui passant en paramètre une chaîne de caractère contenant les informations de la note. Les différents paramètres sont :

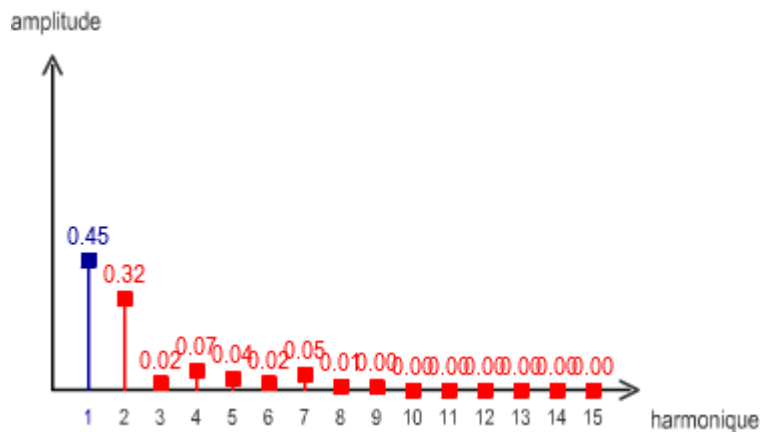
- Le nom de la note (do, ré, mi..., si, do) ;
- Le numéro de l'octave (-1 à 7 pour le piano) ;
- L'altération de la note (dièse, bémol ou 0 pour rien) ;
- La durée de la note (par rapport à la note de référence) ;
- Le coefficient d'amplitude de la note (pour régler le « volume » de la note) ;

Après avoir appelé la fonction, celle-ci s'occupe d'extraire toutes les données citées pour pouvoir créer notamment la fréquence de la note en question. En effet c'est à partir de cette fréquence que la note va pouvoir être créée. Celle-ci est obtenue grâce au lien existant entre les octaves sur un piano, par exemple entre le Do d'une octave et celui de l'octave du dessus la fréquence est multipliée par 2,02.

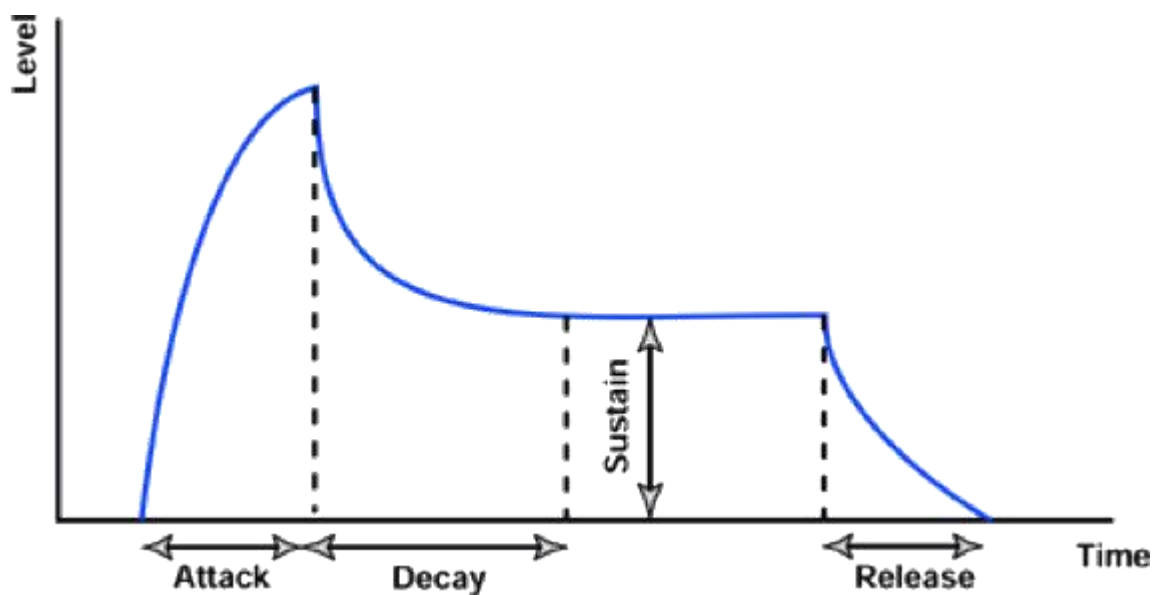
Note/octave	0	1	2	3	4	5	6	7
do ou si#	32,70	65,41	130,81	261,63	523,25	1046,50	2093,00	4186,01
do# ou réb	34,65	69,30	138,59	277,18	554,37	1108,73	2217,46	4434,92
ré	36,71	73,42	146,83	293,66	587,33	1174,66	2349,32	4698,64
ré# ou mib	38,89	77,78	155,56	311,13	622,25	1244,51	2489,02	4978,03
mi ou fab	41,20	82,41	164,81	329,63	659,26	1318,51	2637,02	5274,04
fa ou mi#	43,65	87,31	174,61	349,23	698,46	1396,91	2793,83	5587,65
fa# ou solb	46,25	92,50	185,00	369,99	739,99	1479,98	2959,96	5919,91
sol	49,00	98,00	196,00	392,00	783,99	1567,98	3135,96	6271,93
sol# ou lab	51,91	103,83	207,65	415,30	830,61	1661,22	3322,44	6644,88
la	55,00	110,00	220,00	440,00	880,00	1760,00	3520,00	7040,00
la# ou sib	58,27	116,54	233,08	466,16	932,33	1864,66	3729,31	7458,62
si ou dob	61,74	123,47	246,94	493,88	987,77	1975,53	3951,07	7902,13

2) Création de la note (« synthad_enveloppe.m » ; « adsr_gen.m ») :

Cette fonction est en grosse partie basée sur celle vue en cours et elle permet la création du signal pour chaque note. Dans cette fonction un signal sinusoïdal est créé en fonction de la fréquence de la note, pour se rapprocher le plus possible du piano, qui possède un « son » particulier caractérisé par des harmoniques spéciales, ce n'est pas une sinusoïde mais un ensemble de sinusoïde qui est créé pour chaque harmonique et chaque fréquence. Ci-dessous les harmoniques pour le piano avec les fréquences correspondantes.



De plus, nous avons ajouté une enveloppe à ce signal, cette enveloppe va forcer la sinusoïde à prendre une certaine forme caractéristique au piano toujours. Pour cela nous avons utilisé la méthode dite de l'ADSR (Attack, Decay, Sustain, Release) qui décrit la manière dont l'amplitude varie au cours du temps après avoir joué une note. Dans un premier temps la phase d'attaque du piano est très courte c'est-à-dire qu'après avoir joué la note celle-ci atteint son amplitude maximale quasi instantanément, ensuite il y a la phase où l'amplitude décroît jusqu'à atteindre la phase de maintien de la note qui dure plus ou moins longtemps en fonction de la durée voulue. Enfin la note est relâchée et l'amplitude converge vers 0.

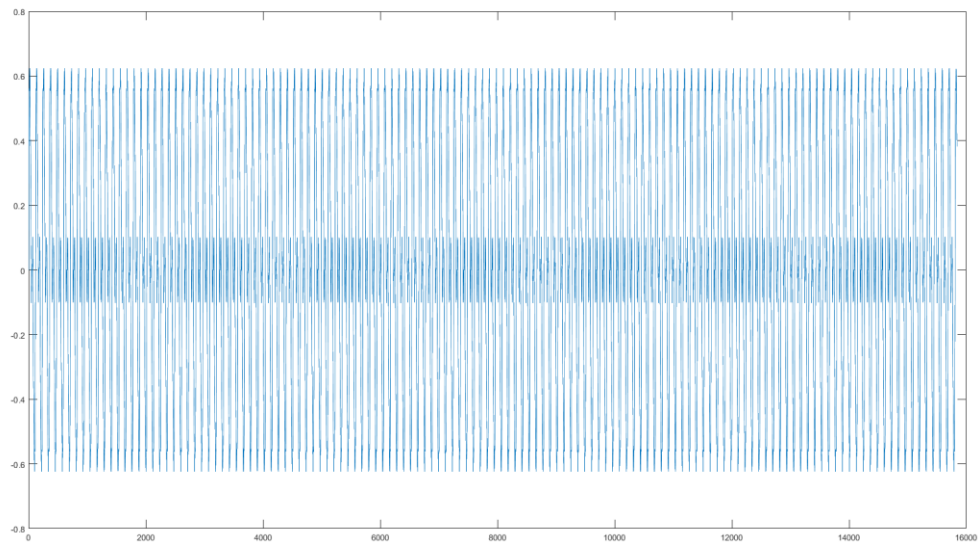


- [Qu'est-ce que l'ADSR ? \(anglais\)](#)

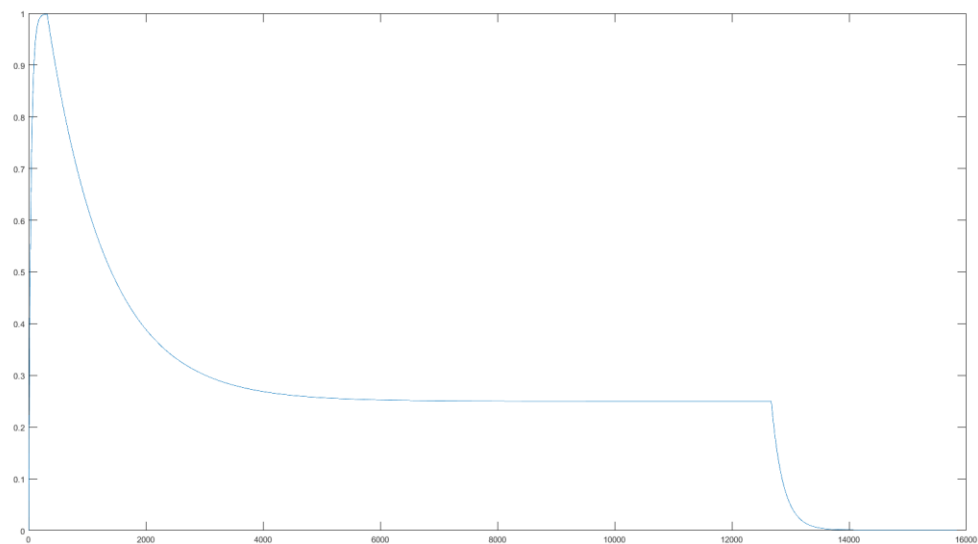
Pour créer le vecteur que l'on allait multiplier à la sinusoïde terme à terme, nous nous sommes inspiré de la fonction présentée dans le document suivant que nous avons légèrement modifiée pour obtenir un résultat optimal.

Nous avons notamment modifié les durées de chaque phase ainsi que les gains qui correspondent aux coefficients de courbe de chaque phase. En jouant avec ces différents paramètres on a pu obtenir une enveloppe comme ci-dessous :

➤ Signal de base (sans enveloppe)

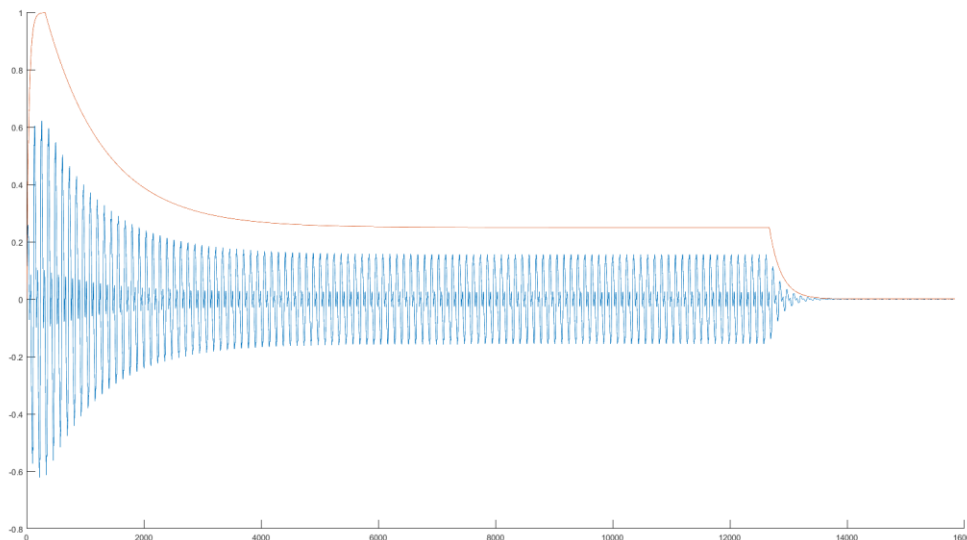


➤ Vecteur de l'enveloppe



➤ [Description d'une enveloppe piano sous Matlab](#)

➤ Signal avec l'enveloppe d'appliquée

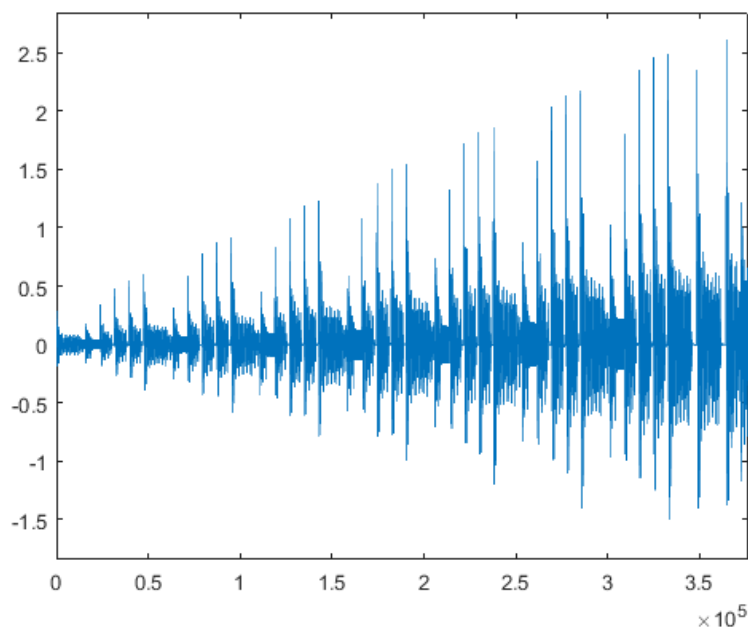


3) Création des vecteurs de notes et lecture (« GoT.m ») :

Enfin après avoir créé l'ensemble de nos notes nous avons dû les mettre bout à bout dans un vecteur que l'on passera dans la fonction propre à Matlab pour lire la musique. Pour des soucis d'ergonomie et de lecture nous avons divisé ce vecteur en plusieurs parties (chaque sous vecteur correspond à une ligne de la partition soit deux mesures, la main droite et la main gauche étant dissociées), et nous avons également créé des variables intermédiaires pour les suites de notes se répétant plusieurs fois.

À noter que nous avons également ajouté le crescendo et le decrescendo au début et à la fin de la musique en utilisant un vecteur, de la même taille que celui sur lequel on veut appliquer l'effet, contenant des valeurs croissantes ou décroissantes comprises entre 0,1 et 1, que l'on va multiplier aux signaux pour les coefficienter et donc les amplifier ou non.

➤ Crescendo du début :



Finalement après avoir obtenu nos vecteurs nous avons sommé celui de la main droite et celui de la main gauche si nous voulions un son mono et créé un vecteur 2x1 contenant dans la première ligne le vecteur de la main droite et dans la deuxième ligne le vecteur de la main gauche si nous voulions le son en stéréo (voir la description de la fonction soundsc)

NB : Le fichier GoT.wav n'est pas une retranscription parfaite de la musique écoutée sur Matlab selon moi, je pense que cela est dû à la compression du fichier et je ne sais pas comment la régler donc il est idéal d'écouter le morceau sur Matlab en plus du fichier.

Les différents liens en entier :

- <https://musescore.com/user/17448756/scores/3536451>
- <https://blog.landr.com/wp-content/uploads/2016/10/ASDR-01.jpg>
- http://www.ece.nmsu.edu/~pdeleon/Research/Publications/ASEE_GSW_2000.pdf