Refaire ses gammes (Denise Vella-Chemla, 7.6.2017)

On va étudier les fréquences des notes de la gamme, suite au visionnage de la petite video Science étonnante #41 intitulée *Les mathématiques de la musique* qui se trouve ici https://www.youtube.com/watch?v=cTYvCpLRwao.

David Louapre explique dans cette video pourquoi il y a 12 notes dans la gamme chromatique et pourquoi certaines notes s'accordent bien (les écouter jouées ensemble est agréable à l'oreille).

On retiendra en résumant que la raison essentielle à cela est que $2^{19} = 524288 \simeq 3^{12} = 531441$ et qu'il s'agit d'écarter les notes entre elles en “mettant en face” 12 facteurs 2 ou 4 et 12 facteurs 3, ce qu'on fait en utilisant pour passer d'une note à la suivante immédiate les fractions rationnelles $\left\{\frac{3}{2}, \frac{3}{4}, \frac{3}{4}, \frac{3}{2}, \frac{3}{4}, \frac{3}{2}, \frac{3}{4}, \frac{3}{4}, \frac{3}{2}, \frac{3}{4}, \frac{3}{4}, \frac{3}{2}\right\}$ en partant initialement d'une note *La*.

L'ensemble des dénominateurs comprend 19 facteurs 2, et l'ensemble des numérateurs comprend 12 facteurs 3 et les fractions rationnelles $\frac{3}{2}$ sont équitablement réparties au sein de l'ensemble des fractions $\frac{3}{4}$ plus nombreuses.

Le tableau ci-dessous est celui fourni dans la video : on passe d'une note (de sa fréquence) à celle de la note à sa droite en “passant à la quinte”, en multipliant la fréquence par $\frac{3}{2}$ ou par $\frac{3}{4}$ (on a mis le multiplicateur qui permet de passer des nombres d'une colonne à ceux de la suivante en bas de cette colonne) ; on passe d'une note à celle au-dessous dans la même colonne en “passant à l'octave” (en multipliant la fréquence par 2).

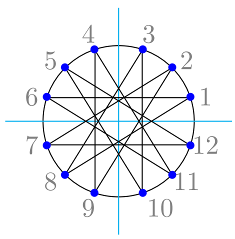
La	*Mi*	*Si*	*Fa#*	*Do#*	*Sol#*	*Ré#*	*La#*	*Fa*	*Do*	*Sol*	*Ré*	~~*La*~~
55	83	62	46	70	52	78	59	44	66	50	37	~~56~~
110	165	124	93	139	104	157	117	88	132	99	74	~~112~~
220	330	248	186	278	209	313	235	176	264	198	149	~~223~~
440	660	495	371	557	418	626	470	352	529	396	297	~~446~~
880	1320	990	743	1114	835	1253	940	705	1057	793	595	~~892~~
1760	2640	1980	1485	2228	1671	2506	1879	1410	2114	1586	1189	~~1784~~
3520	5280	3960	2970	4455	3341	5012	3759	2819	4229	3172	2379	~~3568~~
$\times\frac{3}{2}\nearrow$	$\times\frac{3}{4}\nearrow$	$\times\frac{3}{4}\nearrow$	$\times\frac{3}{2}\nearrow$	$\times\frac{3}{4}\nearrow$	$\times\frac{3}{2}\nearrow$	$\times\frac{3}{4}\nearrow$	$\times\frac{3}{4}\nearrow$	$\times\frac{3}{2}\nearrow$	$\times\frac{3}{4}\nearrow$	$\times\frac{3}{4}\nearrow$	$\times\frac{3}{2}\nearrow$	
8	3	10	5	12	7	2	9	4	11	6	1	

Permutons maintenant les colonnes de ce tableau de façon à remettre les notes dans l'ordre habituel d'une gamme ascendante. On passe cette fois-ci de chaque fréquence à la suivante selon l'ordre habituel de lecture (de gauche à droite et de haut en bas) en multipliant cette fréquence par $\sqrt[12]{2} \simeq 1.059463\ldots$.

1	2	3	4	5	6	7	8	9	10	11	12	
Ré	*Ré#*	*Mi*	*Fa*	*Fa#*	*Sol*	*Sol#*	*La*	*La#*	*Si*	*Do*	*Do#*	
74	78	83	88	93	99	104	110	117	124	132	139	$+4 \to +7$
149	157	165	176	186	198	209	220	235	248	264	278	$+8 \to +14$
297	313	330	352	371	396	418	440	470	495	529	557	$+16 \to +28$
595	626	660	705	743	793	835	880	940	990	1057	1114	$+31 \to +57$
1189	1253	1320	1410	1485	1586	1671	1760	1879	1980	2114	2228	$+64 \to +114$
2379	2506	2640	2819	2970	3172	3341	3520	3759	3960	4229	4455	$+127 \to +226$
4758	5012	5280	5638	5940	6344	6682	7040	7518	7920	8458	8910	$+254 \to +452$

On peut calculer approximativement par ligne les additions à effectuer pour passer d'une note à la suivante. On les a notées en fin de lignes en gris, de la plus petite somme à effectuer à la plus grande.

On peut enfin en ayant numéroté les colonnes de 1 à 12, noter ces nombres sur un cercle pour bien montrer la cyclicité et relier les notes successives par “passage à la quinte” (on a reporté ces nombres dans la dernière ligne du premier tableau également). Du fait de la cyclicité, ce dessin possède de multiples symétries.



Ce qu'il semble intéressant de faire, c'est de prolonger notre tableau de fréquences vers le domaine de l'inaudible, en réappliquant vers le haut du tableau les divisions par 2 des nombres des colonnes : moyennant le passage du discret au continu, on a un rapprochement des valeurs qu'il faut garder à l'esprit, en se rappelant cependant que toutes les fréquences fournies dans les 2 tableaux sont des valeurs entières arrondies les plus proches de réels.

Ré	*Ré#*	*Mi*	*Fa*	*Fa#*	*Sol*	*Sol#*	*La*	*La#*	*Si*	*Do*	*Do#*
1	1	1	1	1	2	2	2	2	2	2	2
2	2	3	3	3	3	3	3	4	4	4	4
5	5	5	6	6	6	7	7	7	8	8	9
9	10	10	11	12	12	13	14	15	16	17	18
19	20	21	22	23	25	26	28	29	31	33	35
37	39	42	44	47	50	52	55	59	62	66	70
74	78	83	88	93	99	104	110	117	124	132	139

On pourrait de la même manière fabriquer une gamme à 3 notes : on utilise le fait que $2^7 = 128 \simeq 5^3 = 125$. On doit mettre 7 facteurs 2 en face de 3 facteurs 5. On utilise les 3 fractions rationnelles $\left\{\frac{5}{4}, \frac{5}{8}, \frac{5}{4}\right\}$.

Admettons qu'on parte du nombre 500. Multiplié par $\frac{5}{4}$, on obtient 625, qu'on multiplie par $\frac{5}{8}$ pour obtenir 391, qu'on multiplie quant à lui par $\frac{5}{4}$ pour obtenir 489. On est quasiment revenu au chiffre initial 500.

On remet les nombres dans l'ordre $391 \to 489 \to 500 \to 625$. On doit passer de l'un à l'autre par multiplication par $\sqrt[3]{2} \simeq 1.025992\ldots$ (la suite obtenue est $391, 493, 621$. Il manque un nombre de la séquence, le 500 qui a disparu...?).

On a également $7^5 \simeq 2^{14} \simeq 11^4$ et $13^4 \simeq 2^{15}$ mais les écarts sont de plus en plus grands : $2^{14} = 16384$, $7^5 = 16807$, $11^4 = 14641$ d'une part, et $2^{15} = 32768$ et $13^4 = 28561$ d'autre part, "presque" égaux.

$\sqrt[4]{2} \simeq 1.18921\ldots$
$\sqrt[14]{11} \simeq 1.18682\ldots$
$\sqrt[15]{13} \simeq 1.18649\ldots$

$\sqrt[5]{2} \simeq 1.1487\ldots$
$\sqrt[14]{7} \simeq 1.14912\ldots$

Ci-dessous, un programme pour convertir les fréquences des notes de la gamme en fixant l'origine 0 pour la plus petite fréquence (32.7) trouvée dans un tableau sur la toile (cf. code) en nombres espacés de 1. Pour convertir une fréquence, on soustrait $\ln(32.7)$ à son logarithme népérien, on divise par $\ln(2)$ et on multiplie par 12.

```
\end{document}
#include <iostream>
#include <stdio.h>
#include <cmath>
#include <math.h>
#include <string>

float convertis(float x, float y) {
  float calcul ;

  calcul = (12*(log(y)-log(x)))/log(2) ;
  return calcul ;
}

int main (int argc, char* argv[]) {
  float freq[13][10] ;
  float freq2[13][10] ;
  float freq3[13][10] ;
  std::string trad[13] ;
  int i, j ;
  float rac12emede2=1.059463 ;
  float prems, convprems, calcul ;

  freq[1][0] = 32.7 ; // do
  freq[2][0] = 34.7 ; // do dièse
  freq[3][0] = 36.7 ; // ré
  freq[4][0] = 38.9 ; // ré dièse
  freq[5][0] = 41.2 ; // mi
  freq[6][0] = 43.7 ; // fa
  freq[7][0] = 46.3 ; // fa dièse
  freq[8][0] = 49.0 ; // sol
  freq[9][0] = 51.9; // sol dièse
  freq[10][0] = 55.0; // la
  freq[11][0] = 58.3 ; // la dièse
  freq[12][0] = 61.7 ; // si

  freq[1][1] = 65.4 ; // do
  freq[2][1] = 69.3 ; // do dièse
  freq[3][1] = 73.4 ; // ré
  freq[4][1] = 77.8 ; // ré dièse
  freq[5][1] = 82.4 ; // mi
  freq[6][1] = 87.3 ; // fa
  freq[7][1] = 92.5 ; // fa dièse
  freq[8][1] = 98.0 ; // sol
  freq[9][1] = 103.8 ; // sol dièse
  freq[10][1] = 110.0 ; // la
  freq[11][1] = 116.5 ; // la dièse
  freq[12][1] = 123.5 ; // si

  freq[1][2] = 130.8 ; // do
  freq[2][2] = 138.6 ; // do dièse
  freq[3][2] = 146.8 ; // ré
  freq[4][2] = 155.6 ; // ré dièse
  freq[5][2] = 164.8 ; // mi
  freq[6][2] = 174.6 ; // fa
  freq[7][2] = 185.0 ; // fa dièse
  freq[8][2] = 196.0 ; // sol
  freq[9][2] = 207.7 ; // sol dièse
  freq[10][2] = 220.0 ; // la
  freq[11][2] = 233.1 ; // la dièse
  freq[12][2] = 246.9 ; // si
```

```
freq[1][3] = 261.6 ; // do
freq[2][3] = 277.2 ; // do dièse
freq[3][3] = 293.7 ; // ré
freq[4][3] = 311.1 ; // ré dièse
freq[5][3] = 329.6 ; // mi
freq[6][3] = 349.2 ; // fa
freq[7][3] = 370.0 ; // fa dièse
freq[8][3] = 392.0 ; // sol
freq[9][3] = 415.3 ; // sol dièse
freq[10][3] = 440.0 ; // la
freq[11][3] = 466.2 ; // la dièse
freq[12][3] = 493.9 ; // si

freq[1][4] = 523.3 ; // do
freq[2][4] = 554.4 ; // do dièse
freq[3][4] = 587.3 ; // ré
freq[4][4] = 622.3 ; // ré dièse
freq[5][4] = 659.3 ; // mi
freq[6][4] = 698.5 ; // fa
freq[7][4] = 740.0 ; // fa dièse
freq[8][4] = 784.0 ; // sol
freq[9][4] = 830.6 ; // sol dièse
freq[10][4] = 880.0 ; // la
freq[11][4] = 932.3 ; // la dièse
freq[12][4] = 987.8 ; // si

freq[1][5] = 1046.5 ; // do
freq[2][5] = 1108.7 ; // do dièse
freq[3][5] = 1174.7 ; // ré
freq[4][5] = 1244.5 ; // ré dièse
freq[5][5] = 1318.5 ; // mi
freq[6][5] = 1396.9 ; // fa
freq[7][5] = 1480.0 ; // fa dièse
freq[8][5] = 1568.0 ; // sol
freq[9][5] = 1661.2 ; // sol dièse
freq[10][5] = 1760.0 ; // la
freq[11][5] = 1864.7 ; // la dièse
freq[12][5] = 1975.5 ; // si

freq[1][6] = 2093.0 ; // do
freq[2][6] = 2217.5 ; // do dièse
freq[3][6] = 2349.3 ; // ré
freq[4][6] = 2489.0 ; // ré dièse
freq[5][6] = 2637.0 ; // mi
freq[6][6] = 2793.8 ; // fa
freq[7][6] = 2960.0 ; // fa dièse
freq[8][6] = 3136.0 ; // sol
freq[9][6] = 3322.4 ; // sol dièse
freq[10][6] = 3520.0 ; // la
freq[11][6] = 3729.3 ; // la dièse
freq[12][6] = 3951.1 ; // si

freq[1][7] = 4186.0 ; // do
freq[2][7] = 4434.9 ; // do dièse
freq[3][7] = 4698.6 ; // ré
freq[4][7] = 4978.0 ; // ré dièse
freq[5][7] = 5274.0 ; // mi
freq[6][7] = 5587.7 ; // fa
freq[7][7] = 5919.9 ; // fa dièse
freq[8][7] = 6271.9 ; // sol
freq[9][7] = 6644.9 ; // sol dièse
freq[10][7] = 7040.0 ; // la
freq[11][7] = 7458.6 ; // la dièse
freq[12][7] = 7902.1 ; // si
```

```
freq[1][8] = 8372.0 ; // do
freq[2][8] = 8869.8 ; // do dièse
freq[3][8] = 9397.3 ; // ré
freq[4][8] = 9956.1 ; // ré dièse
freq[5][8] = 10548.1 ; // mi
freq[6][8] = 11175.3 ; // fa
freq[7][8] = 11839.8 ; // fa dièse
freq[8][8] = 12543.9 ; // sol
freq[9][8] = 13289.8 ; // sol dièse
freq[10][8] = 14080.0 ; // la
freq[11][8] = 14917.2 ; // la dièse
freq[12][8] = 15804.3 ; // si


trad[1] = "do " ;
trad[2] = "do# " ;
trad[3] = "re " ;
trad[4] = "re# " ;
trad[5] = "mi " ;
trad[6] = "fa " ;
trad[7] = "sol " ;
trad[8] = "sol#" ;
trad[9] = "la " ;
trad[10] = "la# " ;
trad[11] = "si " ;
trad[12] = "do " ;

std::cout << "     " ;
for (i = 1 ; i <= 12 ; ++i)
  std::cout << trad[i] << "   " ;
std::cout << "\n" ;

for (j = 0 ; j <= 8 ; ++j)
  {
    for (i = 1 ; i <= 12 ; ++i)
      printf("%10.3f", freq[i][j]) ;
    std::cout << "\n" ;
  }
std::cout << "\n" ;
prems = 32.7 ;
convprems = 0.0 ;
for (j = 0 ; j <= 8 ; ++j)
  {
    for (i = 1 ; i <= 12 ; ++i)
      {
        freq2[i][j] = prems ;
        printf("%10.3f", freq2[i][j]) ;
        prems = prems * rac12emede2 ;
        //convprems = convprems+0.5 ;
        freq3[i][j] = convprems ;
        convprems = convprems+1.0 ;
      }
    std::cout << "\n" ;
  }
std::cout << "\n" ;
//for (j = 0 ; j <= 8 ; ++j)
// {
//   for (i = 1 ; i <= 12 ; ++i)
//    printf("%10.3f", freq3[i][j]) ;
//   std::cout << "\n" ;
// }
```

```
  calcul = (12*(log(1479.826)-log(32.7)))/log(2) ;
  std::cout << calcul << "\n" ;

  for (j = 0 ; j <= 8 ; ++j)
    {
      for (i = 1 ; i <= 12 ; ++i)
        {
          calcul = convertis(32.7, freq2[i][j]) ;
          printf("%9.3f",calcul) ;
        }
      std::cout << "\n" ;
    }
}
```

Résultat de l'exécution

```
     do       do#      re       re#      mi       fa       fa#      sol      sol#     la       la#      si
   32.700   34.700   36.700   38.900   41.200   43.700   46.300   49.000   51.900   55.000   58.300   61.700
   65.400   69.300   73.400   77.800   82.400   87.300   92.500   98.000  103.800  110.000  116.500  123.500
  130.800  138.600  146.800  155.600  164.800  174.600  185.000  196.000  207.700  220.000  233.100  246.900
  261.600  277.200  293.700  311.100  329.600  349.200  370.000  392.000  415.300  440.000  466.200  493.900
  523.300  554.400  587.300  622.300  659.300  698.500  740.000  784.000  830.600  880.000  932.300  987.800
 1046.500 1108.700 1174.700 1244.500 1318.500 1396.900 1480.000 1568.000 1661.200 1760.000 1864.700 1975.500
 2093.000 2217.500 2349.300 2489.000 2637.000 2793.800 2960.000 3136.000 3322.400 3520.000 3729.300 3951.100
 4186.000 4434.900 4698.600 4978.000 5274.000 5587.700 5919.900 6271.900 6644.900 7040.000 7458.600 7902.100
 8372.000 8869.800 9397.300 9956.100 10548.100 11175.300 11839.800 12543.900 13289.800 14080.000 14917.200 15804.300

   32.700   34.644   36.705   38.887   41.199   43.649   46.245   48.995   51.908   54.995   58.265   61.729
   65.400   69.289   73.409   77.774   82.399   87.298   92.489   97.989  103.816  109.989  116.529  123.459
  130.800  138.578  146.818  155.548  164.797  174.597  184.979  195.978  207.632  219.978  233.059  246.917
  261.599  277.155  293.635  311.096  329.594  349.193  369.957  391.956  415.263  439.956  466.117  493.833
  523.198  554.309  587.270  622.191  659.188  698.386  739.914  783.911  830.525  879.911  932.233  987.666
 1046.396 1108.617 1174.539 1244.381 1318.375 1396.770 1479.826 1567.821 1661.049 1759.820 1864.464 1975.331
 2092.790 2217.233 2349.077 2488.760 2636.749 2793.538 2959.651 3135.640 3322.095 3519.637 3728.925 3950.658
 4185.576 4434.463 4698.150 4977.516 5273.494 5587.072 5919.296 6271.276 6644.185 7039.269 7457.845 7901.311
 8371.146 8868.920 9396.293 9955.025 10546.981 11174.137 11838.585 12542.543 13288.360 14078.526 14915.678 15802.609

65.9999
```

```
   0.000    1.000    2.000    3.000    4.000    5.000    6.000    7.000    8.000    9.000   10.000
   11.000
  12.000   13.000   14.000   15.000   16.000   17.000   18.000   19.000   20.000   21.000   22.000
  23.000
  24.000   25.000   26.000   27.000   28.000   29.000   30.000   31.000   32.000   33.000   34.000
  35.000
  36.000   37.000   38.000   39.000   40.000   41.000   42.000   43.000   44.000   45.000   46.000
  47.000
  48.000   49.000   50.000   51.000   52.000   53.000   54.000   55.000   56.000   57.000   58.000
  59.000
  60.000   61.000   62.000   63.000   64.000   65.000   66.000   67.000   68.000   69.000   70.000
  71.000
  72.000   73.000   74.000   75.000   76.000   77.000   78.000   79.000   80.000   81.000   82.000
  83.000
  84.000   85.000   86.000   87.000   88.000   89.000   90.000   91.000   92.000   93.000   94.000
  95.000
  96.000   97.000   98.000   99.000  100.000  101.000  102.000  103.000  104.000  105.000  106.000
  107.000
```

On a ainsi trouvé une fonction qui, grosso-modo, convertit les notes de la gamme tempérée en la suite des entiers (c'est la fonction $f(y) = 12 \times (ln(y) - ln(32.7))/ln(2)$ qui est codée par la fonction $convertis(x, y)$ dans le programme ci-dessus).

On découvre, et c'est surprenant, qu'en calculant les rapports des carrés des parties imaginaires des premiers zéros de ζ, on aboutit aussi à une séquence d'entiers qui ressemble à la suite des entiers successifs, du moins au tout début.

```
zeros[1] = 14.1347
zeros[2]^2/zeros[1]^2 -> 2.21195
zeros[3]^2/zeros[1]^2 -> 3.131
zeros[4]^2/zeros[1]^2 -> 4.63322
zeros[5]^2/zeros[1]^2 -> 5.42928
zeros[6]^2/zeros[1]^2 -> 7.07101
zeros[7]^2/zeros[1]^2 -> 8.38049
zeros[8]^2/zeros[1]^2 -> 9.39602
zeros[9]^2/zeros[1]^2 -> 11.5346
zeros[10]^2/zeros[1]^2 -> 12.4002
zeros[11]^2/zeros[1]^2 -> 14.044
zeros[12]^2/zeros[1]^2 -> 15.9476
zeros[13]^2/zeros[1]^2 -> 17.6288
zeros[14]^2/zeros[1]^2 -> 18.5219
zeros[15]^2/zeros[1]^2 -> 21.2205
zeros[16]^2/zeros[1]^2 -> 22.5221
zeros[17]^2/zeros[1]^2 -> 24.2089
zeros[18]^2/zeros[1]^2 -> 25.9956
zeros[19]^2/zeros[1]^2 -> 28.6861
zeros[20]^2/zeros[1]^2 -> 29.7878
zeros[21]^2/zeros[1]^2 -> 31.5051
zeros[22]^2/zeros[1]^2 -> 34.4067
zeros[23]^2/zeros[1]^2 -> 35.9382
zeros[24]^2/zeros[1]^2 -> 38.256
zeros[25]^2/zeros[1]^2 -> 39.4767
zeros[26]^2/zeros[1]^2 -> 39.4767
```