

Mini Project: 计算机生成和播放音乐

1. Objective

音乐文件有很多格式，但是其主要的数据（Data）部分，保存的都是一个 $M * N$ 的矩阵（其中， N 为通道数）。因此，对于一个乐谱，我们可以用计算机程序生成这样一个矩阵，并用计算机播放出来。本项目的目标就是：

1. 理解音乐的基本要素，比如音调、节拍、音色、基频、和弦等，
2. 理解这些音乐要素如何反映在我们的数据中，
3. 通过 MATLAB 程序，将一张简谱转化成一段可以播放成音乐的数据，
4. 根据数据，产生可播放的音乐，
5. 根据数据，产生模仿某种乐器的音乐

2. 数字简谱

数字简谱是简易的记谱法，用基本音符 1、2、3、4、5、6、7 代表音阶中的 7 个基本级，读音为 do、re、mi、fa、sol、la、ti（中国为 si），英文由 C、D、E、F、G、A、B 表示，休止以 0 表示。图 1 为《天空之城》数字简谱。简谱中主要包含了数字音符和一些其他符号，比如点和横线，这些数字配合上符号决定了每个音的频率和持续时间长度，我们需要通过编码将其输入到程序中，并用来控制音乐的播放。

天空之城

1=D $\frac{4}{4}$

$\dot{1}$ $\underline{6\ 7}$ \parallel $\dot{1}$ $\underline{7\ 1}$ $\dot{3}$ | $7 - - \underline{3\ 3}$ | $\widehat{6\ 5}$ $6\ \dot{1}$ | $5 - \underline{0\ 3\ 3}$ | $4\ \underline{3}$ $\underline{4}$ $\dot{1}$
 $| 3 - \underline{0\ \dot{1}\ \dot{1}\ \dot{1}}$ | $7\ \underline{\sharp 4\ \sharp 4}$ 7 | $7 - 0\ \underline{6\ 7}$ | $\dot{1}$ $\underline{7\ 1}$ $\dot{3}$ | $7 - 0\ \underline{3\ 3}$ | $6\ \underline{5}$ $6\ \dot{1}$
 $| 5 - - \underline{0\ 3}$ | $4\ \underline{1\ 7}$ $\underline{7\ 1}$ | $\underline{2\ 2\ 3}$ $\dot{1}\ 0$ | $\underline{1\ 7}$ $\underline{6\ 6\ 7\ 5}$ \sharp \parallel $6 - 0\ \underline{1\ 2}$ | $\underline{3\ 2}$ $\underline{3\ 5}$
 $| 2 - 0\ \underline{5\ 5}$ | $\underline{1\ 7}$ $\dot{1}\ \dot{3}$ | $\underline{3 - 0\ 0}$ | $\underline{6\ 7\ 1\ 7\ 2\ 2}$ | $\dot{1}$ $\underline{5}$ $5\ 0$ | $\underline{4\ 3}$ $\underline{2\ 1}$
 $| \underline{3 - - -}$ | $\underline{3 - 0\ 3}$ | $\underline{6 - 5\ 5}$ | $\underline{3\ 2}$ $\dot{1}$ $\underline{0\ 1}$ | $\underline{2\ 1\ 2}$ $\underline{2\ 5}$ | $\underline{3 - 0\ 3}$
 $| \underline{6 - 5 -}$ | $\underline{3\ 2}$ $\dot{1}$ $\underline{0\ 1}$ | $\underline{2\ 1\ 2}$ $\underline{2\ 7}$ | $6 - 0\ \underline{6\ 7}$ \parallel $6 - - -$ |

Fine

图 1. 《天空之城》 数字简谱

2.1 音调

简谱中 1、2、3、4、5、6、7 (C、D、E、F、G、A、B) 表示七种高低不同的音，我们称之为音调。之所以会呈现不同音调，是因为他们分别对应不同频率的音波。表 1 为 C 调音符与频率对应表。

表 1. C 调中音符与频率(取整后)对应表，单位为 Hz

	1 (C)	2(D)	3(E)	4(F)	5(G)	6(A)	7(B)
低音	262	294	330	349	392	440	494
中音	523	587	659	698	784	880	988
高音	1046	1175	1318	1397	1568	1760	1976
相邻音符 频率比 $n + 1/n$	1.12	1.12	1.057	1.12	1.12	1.12	1.058

从表 1 可以看出，低音频率为同一音符中音频率的一半，同样的，中音频率为同一音符高音频率的一半。也就是说，对应的中音音符和低音音符之间，频率减半，而音程相差一个八度。记在简谱基本音符号下面的小圆点，叫低音点，它表示将基本音符降低一个音组，即降低一个纯八度。记两个圆点表示将基本音符降低两个音组，即降低两个纯八度。记在简谱基本音符号上面的小圆点，叫高音点，它表示将基本音符升高一个音组，即升高一个纯八度。记两个圆点，表示升高两个音组，即升高两个纯八度。

同时，稍加计算，也可以发现，相邻音符之间的频率比，2:1, 3:2, 5:4, 6:5, 7:6 的频率比值均为 $1.12(2^{\frac{1}{6}})$ 左右，而，4:3, 11:7 则为 $1.057(2^{\frac{1}{12}})$ 左右。由于相邻之间音符的频率比值不完全相等，要通过一个音符（比如 1）来计算其他音符的频率有点困难，不过，进一步分析， $\frac{1}{1} = \frac{1}{7} \times \frac{7}{6} \times \frac{6}{5} \times \frac{5}{4} \times \frac{4}{3} \times \frac{3}{2} \times \frac{2}{1} = 2^{\frac{12}{12}} = 2$ ，这也符合上面介绍的中音频率与低音频率之间的比值规律。因此，可以将一组音划分成十二等分，使得相邻音符的频率比值为相等，即 $2^{\frac{1}{12}}$ ，这样间隔 12 个音符以后频率翻倍，这种音律叫十二平均律，亦称“十二等程律”，

其将世界上通用的一组音（八度）分成十二个半音音程（ $2^{\frac{1}{12}}$ ），因此，如果你知道了一组音中一个音符的频率（比如 1），那你就能计算出这一组音中其他音符的频率，在结合间隔八度相差一倍/一半的规律，你能计算出简谱中其他所有音符的频率。图 2 举例说明了如何用十二平均律来计算一组音中所有音阶的频率。图 2 的例子中，主音设定为 440Hz，以此为基准来计算 12 等份的其他音调。主音在乐理中有特别的含义，但在这里，我们当前只需要简单的把它理解为一个基准，更具体的我们将在下一部分解释。而对应到基本音级中，我们可以发现，3 - 4 和 7 - 1i 之间的间隔是一个半音，而其它相邻两个音之间的间隔都是全音（2 个半音），因此，我们也可以通过其中一个音符的频率来计算出其他音符的频率。

此外，简谱中还有升（降）调的符号#（b）号，升调表示比原音级高 1 个半音，降调同理，加上升降调的话，7 个音级就扩展到了十二平均律中的十二个音级，以 C 调为例，如图 3 所示。

总结一下，本节主要介绍在知道一个音符频率下，如何计算其他音符的频率。在十二平均律下，所有音符之间的频率比值相等，因此可以通过这个倍数来计算。在 1-7 的七个基本音级中，3 - 4、7 - 1i 的间隔是半音（ $2^{\frac{1}{12}}$ ），而其它相邻两个音的间隔都是全音（2 个半音）（ $2^{\frac{1}{6}}$ ），因此我们只需要知道其他音符对应于已知音符的位置关系，就可以通过这个频率间隔关系来计算。此外，如果相差了若干个八度，则应先计算



同一组七个音符的频率，再按照相差一个八度，频率翻倍或减半来计算。此外，简谱上出现的数字 1 - 7 加上升降调就有了 12 种音符，再加上上下的小圆点，就有了 $12 \times N$ (N 个八度) 种音符。因此在对简谱编码时，就要考虑用一种合适的方式将这 $12 \times N$ 个音符输入进去。比如，用 3 个变量，分别控制 1 - 7 的基本音级、升降调、相差多少个八度；或者用 1 - 12 表示 1 - 7 的基本音级加上升降调，13 - 24 表示高一个八度， $\square 12 - \square 1$ 表示低一个八度。此外还有考虑停顿 (0)。对应不同的编码方式需要用不同的方式将简谱输入。每个人可以采用自己喜欢的方式进行编码。但是，在接下来的练习中，请务必在每个函数中提供注释介绍你的编码方式。

十二平均律表

将主音设为 a1(440Hz)，来计算所有音的频率，结果如下 (为计算过程更清晰，分数不进行约分)：

音程名称	间隔半音数	十二平均律的倍数	频率
纯一度 (A ¹)	0	$2^0 = 1$	$440 \times 1 = 440$
增一度/ 小二度 (A [#] /B ^b)	1	$\sqrt[12]{2} = 2^{\frac{1}{12}} \approx 1.0594630943592952645618252949463$	$440 \times 2^{\frac{1}{12}} \approx 466.1637615180899164072031297762$
大二度 (B ¹)	2	$\sqrt[6]{2} = 2^{\frac{2}{12}} \approx 1.1224620483093729814335330496792$	$440 \times 2^{\frac{2}{12}} \approx 493.8833012561241118307545418586$
小三度 (C)	3	$\sqrt[4]{2} = 2^{\frac{3}{12}} \approx 1.1892071150027210667174999705605$	$440 \times 2^{\frac{3}{12}} \approx 523.2511306011972693556999870466$
大三度 (C [#])	4	$\sqrt[3]{2} = 2^{\frac{4}{12}} \approx 1.2599210498948731647672106072782$	$440 \times 2^{\frac{4}{12}} \approx 554.3652619537441924975726672023$
纯四度 (D)	5	$\sqrt[12]{32} = 2^{\frac{5}{12}} \approx 1.3348398541700343648308318811845$	$440 \times 2^{\frac{5}{12}} \approx 587.3295358348151205255660277209$
增四度/ 减五度 (D [#] /E ^b)	6	$\sqrt{2} = 2^{\frac{6}{12}} \approx 1.4142135623730950488016887242097$	$440 \times 2^{\frac{6}{12}} \approx 622.2539674441618214727430386522$
纯五度 (E)	7	$\sqrt[12]{128} = 2^{\frac{7}{12}} \approx 1.4983070768766814987992807320298$	$440 \times 2^{\frac{7}{12}} \approx 659.2551138257398594716835220930$
小六度 (F)	8	$\sqrt[3]{4} = 2^{\frac{8}{12}} \approx 1.5874010519681994747517056392723$	$440 \times 2^{\frac{8}{12}} \approx 698.4564628660077688907504812795$
大六度 (F [#])	9	$\sqrt[4]{8} = 2^{\frac{9}{12}} \approx 1.6817928305074290860622509524664$	$440 \times 2^{\frac{9}{12}} \approx 739.9888454232687978673904190852$
小七度 (G)	10	$\sqrt[6]{32} = 2^{\frac{10}{12}} \approx 1.781797436280678609480452411181$	$440 \times 2^{\frac{10}{12}} \approx 783.9908719634985881713990609195$
大七度 (G [#])	11	$\sqrt[12]{2048} = 2^{\frac{11}{12}} \approx 1.8877486253633869932838263133351$	$440 \times 2^{\frac{11}{12}} \approx 830.6093951598902770448835778670$
纯八度 (A)	12	$2^1 = 2$	$440 \times 2 = 880$

其中 $\sqrt[12]{2} = 2^{\frac{1}{12}} \approx 1.0594630943593$

$$\approx \frac{18}{17} = 1.05882 \text{ 99 音分}$$

$$\approx \frac{107}{101} = 1.05941 \text{ 99.9 音分}$$

$$\approx \frac{11011}{10393} = 1.05946310 \text{ 100 音分}$$

图 2. 十二平均律表

C调音符与频率对照表

音符	频率/Hz	音符	频率/Hz	音符	频率/Hz
低音1	262	中音1	523	高音1	1046
低音1#	277	中音1#	554	高音1#	1109
低音2	294	中音2	587	高音2	1175
低音2#	311	中音2#	622	高音2#	1245
低音3	330	中音3	659	高音3	1318
低音4	349	中音4	698	高音4	1397
低音4#	370	中音4#	740	高音4#	1480
低音5	392	中音5	784	高音5	1568
低音5#	415	中音5#	831	高音5#	1661
低音6	440	中音6	880	高音6	1760
低音6#	466	中音6#	932	高音6#	1865
低音7	494	中音7	988	高音7	1976

图 3. C 调音符与频率对应表

MATLAB 练习 1: 编写函数

function freq = tone2freq(tone, noctave, rising)

% tone: 输入数字音符, 数值范围1到7

% noctave: 高或低八度的数量, 数值范围整数。0表示中音, 正数表示高noctave个八度, 负数为低noctave个八度

% rising: 升或降调。1为升, -1为降, 0无升降调

% freq为输出的频率

以 1=440 Hz 为主音, 编写函数, 要求能够输出 1 - 7 音符, 高若干八度, 低若干八度, 以及升降调的频率。

2.2 调号

2.1 中, 我们介绍了在已知 1 个音符的频率的前提下计算其他音符的频率。因此, 我们需要前提知道这个音符的频率。在简谱的开头, 我们会看到 1=C 或 1=D 这样的记号, 这表示该简谱是以 C 调或 D 调来记谱的, 同时也规定了该简谱中 1 的频率 (主音)。C 调或 D 调就是我们一般所谓的调号, 不同调号之间 1 的频率 (主音) 有一定的对应关系, 这个关系我们通过表 2 来介绍。表 2 中, 第一行我们给出了 C 大调下音符 1-7 的频率, 第二行是这七个音符的数字符号和英文符号, 剩下的行给出了 C-B 调中的主音频率。你会发现, C 调 (1=C) 中 1 的频率就是 C 调中 1 (C) 的频率 (近乎于废话), D 调 (1=D) 中 1 的频率就是 C 调中 2 (D) 的频率, E 调 (1=E) 中 1 的频率就是 C 调中 3 (E) 的频率, 以此类推, 也就是说, 只要知道了 C 调的频率, 同一组其他调的主音 1 的频率就可以按照这个音符对应 C 调中 1 的位置关系来计算, 这样我们就能确定其他调的主音频率了。

总结, 本节主要介绍调号。调号确定了该简谱中 1 的频率, 再结合上 2.1 中介绍的规律, 就能算出整张简谱中所有音符的频率。所以, 我们首先需要计算 C 调中 7 个音符的频率, 然后计算其他调中的主音频率 (如果简谱用其他调写的), 最后再计算剩下的其他音符的频率。

表 2 不同调号主音频率，单位 Hz

C 调频率	261.5	293.5	329.5	349	391.5	440	494
C 调音符 (英文名)	1(C)	2(D)	3(E)	4(F)	5(G)	6(A)	7(B)
C 调	1=C (261.5)						
D 调		1=D (293.5)					
E 调			1=E (329.5)				
F 调				1=F (349)			
G 调					1=G (391.5)		
A 调						1=A (440)	
B 调							1=B (494)

MATLAB 练习 2: 在 2.1 函数的基础上，编写函数

function freq = tone2freq(tone, scale, noctave, rising)

% tone为输入数字音符，scale 为调号，noctave 为高低八度数量，rising为升降调，freq为输出的频率，编写函数，要求能够计算出一张简谱中任何一个音符的频率。

2.3 音符长短

简谱中，音的长短是在基本音符的基础上加短横线和/或附点表示的。

1) 短横线的用法有两种：写在基本音符右边的短横线叫增时线。增时线越多，音的时值就越长。

不带增时线的基本音符叫四分音符，每增加一条增时线，表示延长一个四分音符的时间。

写在基本音符下面的短横线叫减时线。减时线越多，音就越短，每增加一条减时线，就表示缩短为原音符音长的一半。

2) 写在音符右边的小圆点叫做附点，表示延长前面音符时值的一半。附点往往用于四分音符和少于四分音符的各种音符。带附点的音符叫附点音符。

在这里，我们也需要考虑编码的问题，如何将每个音符对应的持续时间输入，我们可以以 1s 的单个音符持续时长作为基准，加上一条增时线则时间变为 2，加上一条减时线则时间变为 0.5，小圆点同理。

3. 生成不同频率波形

在 MATLAB 中，生成特定频率的波形一般用三角函数。

例 1:

```
fs = 8192; f = 1; T = 1/f;
```

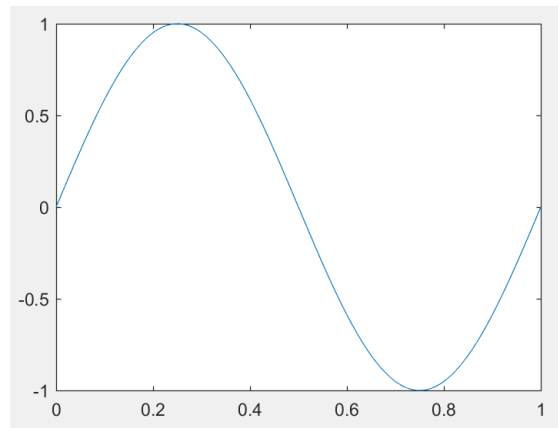
```
t = linspace(0, T, fs);
```

Minor revisions are made with Prof. Yu Yajun's Course Materials. Her contributions are greatly acknowledged.



```
y = sin(2*pi*f*t)
```

以上代码生成了一个频率为 1Hz 的正弦波。



使用 `sound` 函数播放（用 `help` 命令了解 `sound` 函数用法），在例 1 后面添加以下代码：

```
sound(y, fs);
```

你会发现，什么都听不到，因为，人类的听力频率范围是 20-20k Hz. 上面产生的正弦波频率太低了。将例 1 代码改成：

例 2:

```
fs = 8192; f = 440; T = 1/f;
```

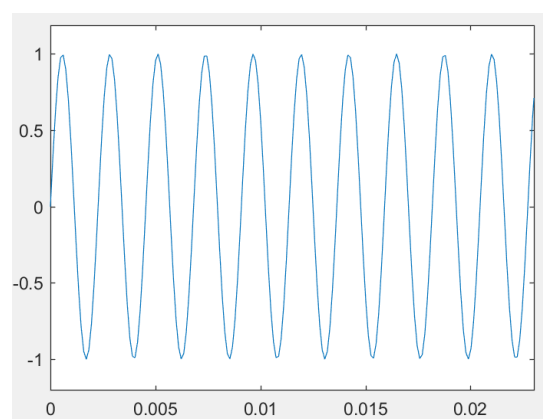
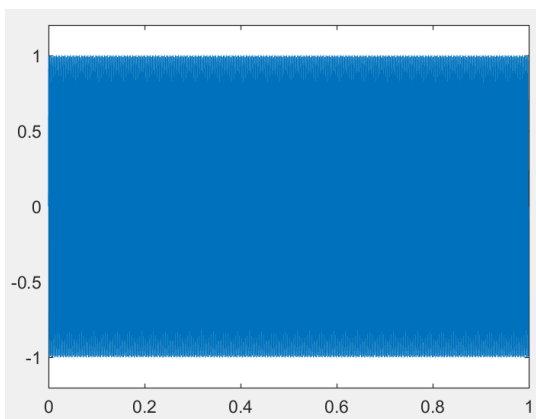
```
t = linspace(0, 1, fs);
```

```
y = sin(2*pi*f*t);
```

```
plot(t,y);
```

```
sound(y, fs);
```

你会听到一个持续 1 秒的‘du’的声音。波形如下图所示（右图为放大后）：



如果需要调整持续时长，则要增加 `t` 的长度，将例 2 第二行改为：

```
rhythm = 5;
```

```
t = linspace(0, rhythm, fs * rhythm);
```

使用 `sound` 函数播放，你会听到一个持续 5 秒的‘du’的声音。2.3 节中介绍的符号决定了音乐中某个音符持续的时长，对应在上面代码中的 `rhythm`。以 4/4 拍（以四分音符为一拍，每小节四拍）为例，每个音符持续

时间设为 1/4 秒（可以自己定义），则延半拍（音符右边带小圆点）共持续 $1/4 + 1/8$ 秒，一条减时线持续 1/8 秒，以此类推。

将数字简谱中的每个音符都转换成 1 - 7 表示的音级（频率），再结合每个音符持续的时长，在使用上面介绍到的代码，为每个音符生成对应的波形，将每个音符对应的波形连接在一起，你将得到一段简单的音乐，你可以使用 sound 函数播放，也可以使用 audiowrite 函数（使用 help 命令了解其用法）将其写入音乐文件中。

MATLAB 练习 3 编写函数

`function waves = gen_wave(tone, scale, noctave, rising, rhythm, fs)`

% tone 为数字音符，scale 为调号，noctave 为高低八度数量，rising 为升降调，rhythm 为节拍，即每个音符持续时长，fs 为采样频率，

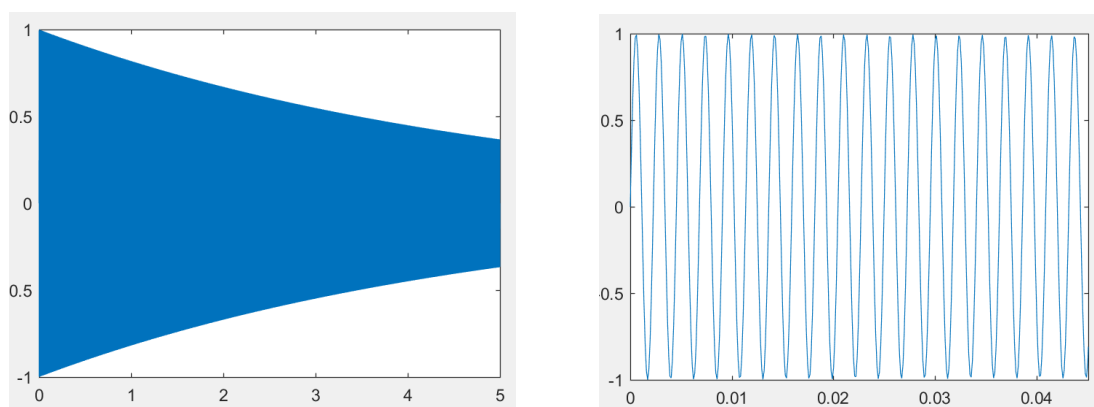
将图 1 中的《天空之城》数字简谱转换成波形文件。

4 音量波动

考虑到乐器演奏时，振动会有衰减，不会以固定幅度持续振动，因此一个包络衰减函数能够更加真实的模拟音乐的产生：

`waves = y.*exp(-x/rhythm));`

在例 2 第三行后面增加这样一行代码，对单个音符的波形进行指数衰减，新的波形如下图所示：



当然，也有很多别的衰减函数或许能带来更加真实的听感，如线性衰减，平方衰减等，甚至可以尝试下其他的波动。同时，这种衰减也可以在一个节拍内的几个音符中进行，比如单个音符的波形加一个衰减函数，几个音符的波形连在一起后再加上一个衰减函数。

MATLAB 练习 4： 尝试使用不同的衰减函数对练习 2 中输出结果进行处理，使用 sound 函数播放并比较其听感，使用 plot 函数画出其波形并分析其播放效果，并选择其中你认为最好的。

5. 泛音/不同乐器的音色区别

前面我们讲的乐谱中的音调都是指音乐的基频。而用乐器演奏音乐时，除了发出乐谱中的基频声音外，由于乐器的发声原理，还产生数量不等的驻波。驻波是指，当一根琴弦两端被固定时，我们拨动琴弦，琴弦振动部分的长度必然是半波长的整数倍，即，发出的声音频率包含基频以及基频的整数倍谐频。

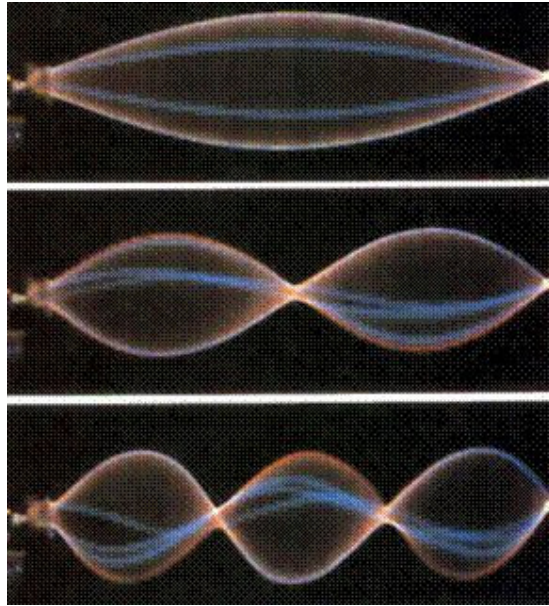
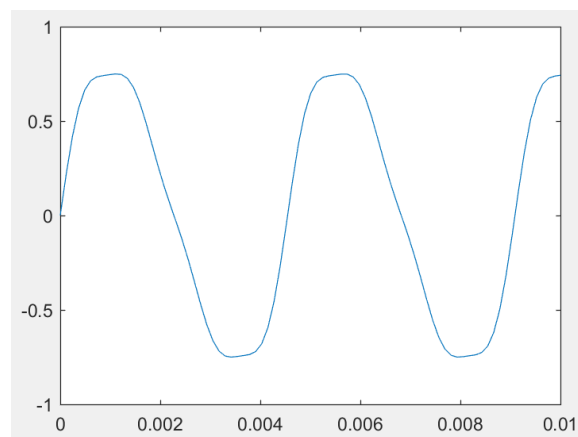


图 5. 驻波原理

所以乐器弹奏时会产生包括基频和若干整数倍频率的谐频，而主要的能量集中于基频。对于 2 倍、3 倍、4 倍、5 倍……的谐频，不同乐器这些谐频的能量比例各不相同。如果我们调整这个比例，将产生音色完全不同的声波：

```
fs = 8192; f = 220; T = 1/f;  
rhythm = 1;  
t = linspace(0, rhythm, fs * rhythm);  
y = 0.8*sin(2*pi*f*t)+0.1*sin(2*pi*2*f*t)+ ...  
0.05*sin(2*pi*3*f*t)+0.05*sin(2*pi*4*f*t);  
%waves = y.*exp(-x/rhythm);  
plot(t,y);  
axis([0 0.01 -1 1]);
```

如果用上面这段程序来产生某个音符的波形，得到的新的波形将如图所示，使用 `sound` 函数播放，你会发现，其音色明显与 2 中不同，将所有音符转化成波形（这里的包络衰减可以保留），再将这些音符连起来，试听以下整段音乐的音质，你会发现，跟单纯的基频生成的音乐有明显区别。





MATLAB 练习 5: 在练习 3 的基础上, 尝试不同的谐波能量比例, 用 `plot` 分析其波形以及频谱波形, 并分析产生的音乐的音色差别。选择你喜欢的音色, 将你喜欢的一首音乐转换成 `wav` 格式的音乐。

6. 实验要求

1. 完成 MATLAB 练习 1-5, 包括代码, 结果, 观察, 分析等。
2. 完成完整的代码 (包含音符频率计算 (`tone2freq`), 单个音符波形数据生成(`gen_wave`), 整个简谱波形数据生成(`gen_music`)共三个函数文件, 以及将波形数据保存成音乐文件 (建议 `wav` 格式) 的脚本文件), 从一份简谱生成一段音乐数据, 尝试调整代码中的包络衰减、谐波能量比例等, 并分析其对听感的影响, 尝试模拟某种乐器的声音, 并给出结果分析。
3. 概括通过这个项目的联系, 你的心得体会、所思所想。
4. 提交文件清单: 1) `pdf` 格式实验报告一份, 包含代码, 图片 (必须) 以及分析过程; 2) 源代码压缩包, 包含上文提到的三个函数和一个脚本文件。简谱数据文件以及最终保存下来的音乐文件。