

1 自作の関数

Matlab でも、自作の関数を作成することができる。Matlab では、原則として、関数は1つのファイルに1つの関数を作り、ファイル名は関数名と同じにしなければならない。関数を定義するファイルを M ファイルとよぶ。

M ファイルを作成するには、[ファイル] メニューの新規作成から [M ファイル] を選択する。

下に .wav ファイルを再生する関数 wavfile_play の例を示す。

```
function [y, fs, nbits] = wavfile_play(filename)
[y, fs, nbits] = wavread(filename);
wavplay(y, fs);
```

関数を作成したら、[ファイル] メニューから保存を選択すると、work フォルダに保存され、Command window から組み込み関数と同様に利用できるようになる。

```
>> wavfile_play('a-.wav');
>>
```

上記の例では、[y, fs, nbits] は出力引数である。一般のプログラミング言語のように return などで明示的に返却値を設定する必要はなく、関数内で値を設定すると、その値が返却される。

2 課題: 簡単な音楽生成プログラム

単音の楽譜情報を与えたら、正弦波でメロディーを生成して再生するプログラムを作成する。

2.1 note

音符情報を与えたら、そのデータを返却する関数 note を作成する。

```
function wave = note(num, dur, fs)
t = 0:1/fs:dur;
freq = (1)
wave = (2)
```

課題 1 (1), (2) を埋めてこの関数を完成させよ。

ただし、num は音階に対応する番号である。中央の C(ド) を 60 とし、440Hz の A (ラ) を 69 とする。dur は音符が続く秒数をあらわす。fs は標本化周波数 (たとえば、16KHz のときには 16000 とする) である。

つまり、下記のように利用することを想定している。

```
>> y = note(60, 1, 16000)
```

中央のドの周波数は次の式で計算できる。

$$f = 440 \times 2^{-9/12} \sim 262(\text{Hz}) \quad (1)$$

wave は、周波数 freq を持つ正弦波の dur 秒間の波形データとなる。

3 課題: 簡単な音楽生成プログラム (2)

楽譜情報に基づき、前節の `note` 関数を利用して楽曲の波形データを生成するスクリプトを考える。

「きらきら星」の冒頭を生成する例を示す。この例では、「ドドソソララソー」の部分生成する。最初の「ド」から「ラ」までは、全て四分音符であるが、最後の「ソー」の部分は二分音符である。プログラムを簡単にするため、まず、四分音符は 1 秒、二分音符は 2 秒であるとする。

このスクリプトでは、`pitch` で、音の高さを指定する。指定方法は、音階を整数で表す。ここでは、中央の C を 60 とする。半音上の C# は 61、D は 62 となる。`duration` で、音の持続時間を秒数であらわす。つまり、`pitch(60)` と `duration(1)` で中央の C が 1 秒間続くことをあらわす。

```
% play music1
pitch = [60 60 67 67 69 69 67]
duration = [1 1 1 1 1 1 2]

fs = 8000;
n1 = 1;
xx = zeros(1,sum(duration)*fs + 1); % 領域の確保
for kk = 1:length(pitch)
    pp = pitch(kk);
    dd = duration(kk);
    tone = (1) % 前回作成した note 関数を用いる
    n2 = n1 + length(tone) - 1;
    xx(n1:n2) = xx(n1:n2) + tone;
    n1 = n2;
end
soundsc(xx, fs);
```

課題 2 (1) の部分を埋めてスクリプトを完成させよ。

上記のスクリプトでは、

```
pitch = [60 60 67 67 69 69 67]
duration = [1 1 1 1 1 1 2]
```

が、楽譜情報をあらわしている。ただ、このスクリプトのままでは、音符の長さが絶対時間であらわされているため、同じ楽譜でテンポが違う演奏を生成できない。そこで、`duration` は四分音符を 1 とした相対的な音符の長さをあらわすこととする。

課題 3 上記のスクリプトを様々な速度で演奏できるように拡張する。そのために、`tempo` という変数を用意する。

この変数には、BPM 値 (四分音符が 1 分間にいくつ演奏されるかを示す値) を設定する。例えば `temp=120` と設定すると、四分音符が 1 分間に 120 となるので、四分音符一つあたりの長さはいくつにしなければならないか? このような考え方に基づいて、`note` 関数に与える引数を工夫せよ。

楽譜情報には休符もある。休符は、`pitch` が 0 で、必要な長さ分 (つまり、四分休符であれば、1) だけ `duration` がある音符だとして表現することにする。

課題 4 `note` 関数を休符にも対応するように修正し、休符を含む演奏を生成して、スクリプトの動作を確認せよ。