

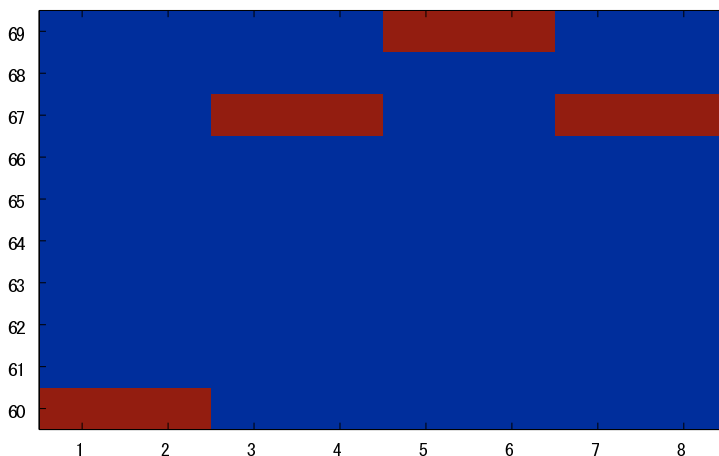
1 ピアノロール

MATLAB など、プログラムによって、楽曲を制作しようと思うと、音符の音高に関する配列、音符の長さに関する配列などを利用するのが一般的であろう。

本格的な制作のためには、パートが増え、曲の長さも長くなる。そのような状態では、配列は全体を閲覧しにくく、入力ミスも発見しにくいという問題がある。

そういった情報を可視化するためには、一般的には楽譜が利用されるので、MATLAB でも楽譜を出力できるようプログラミングすればよいのであるが、楽譜の表現方法は、それほど簡単にプログラミングできるような仕様ではない。

それに対し、多くの DTM ソフトウェアでは、ピアノロール形式を採用している。ピアノロールとは、ピアノの自動演奏用に開発された表現方法であり、例えば、きらきら星の主旋律の冒頭部分は、次のように表現される。



このような表示をプログラミングする方法を考える。

2 imagesc

MATLAB で図を表示するのに便利な関数に `imagesc` がある。

```
>> x=1:2;
>> y=1:2;
>> pix=[0 1; 1 0];
>> imagesc(t,n,pix)
```

`imagesc` 関数はこのように、第 3 引数で指定された行列の情報に応じたビットマップ画像を表示する。

これまでの資料で用いられてきた楽譜情報をあらかず配列に対応したビットマップ画像を表示することを考える。例えば、4 分音符のドミソというフレーズは次のような配列で表現される。

```
>> p=[60 64 67];
>> d=[1 1 1];
```

これと同等の内容を `imagesc` 関数向けに作成することを考える。音高は縦、時間は横になるので、例えば、 a_{11} が 1 番目のド (音高番号 60)、 a_{52} が 2 番目のミ、 a_{83} が 3 番目のソをあらかず行列 A を作ってみる。

```
>> A=[1 0 0;
0 0 0;
0 0 0;
```

```
0 0 0;
0 1 0;
0 0 0;
0 0 0;
0 0 1]
```

```
A =
```

```
1 0 0
0 0 0
0 0 0
0 0 0
0 1 0
0 0 0
0 0 0
0 0 1
```

```
>> x=1:3;
>> y=60:67;
>> imagesc(x,y,A)
```

このままでは、上の方が低い音に対応してしまう。imagesc で表示される図は、左上に (1,1) 座標が存在する。たしかに、行列では、左上が (1,1) となる。しかし、今回の場合、通常の plot と同様に、左下が (0,0) とした方がみやすい。axis 命令で、表示方法を変えることができる。

```
>> axis xy
```

練習 1 きらきら星の冒頭部分 (ドドソソララソ) の部分を 10×8 行列で表現してみよ。また、その行列を用いて上記の例と同じような図を表示せよ。

3 ピアノロール表示プログラム

次のような形で利用できる関数 piano_roll_matrix を考える。

```
>> p1=[60 60 67 67 69 69 67];
>> d1=[1 1 1 1 1 1 2];
>> [pr,t,p]=piano_roll_matrix(p1,d1,60,67,1,1);
>> imagesc(t,p,pr)
>> axis xy;
```

ここで、piano_roll_matrix の第 3 引数、第 4 引数は表示したい音高の最大、最小値、第 5 引数は最小の単位に関する数 (4 分音符を 1 として、最小単位が 8 分音符であれば、2、32 分音符であれば、8 を与える。3 連符がある場合は、最小単位に 3 をかけた数を与える)、第 6 引数は色の指定を行う。(色に関しては、image のヘルプなどを参照のこと。わからない場合は、1 としておけばよい)

```
function [ pr, t, p] = piano_roll_matrix( pitch, duration, pmin, pmax, unit, col )
```

```
t = 1:sum(duration);
duration = unit * duration;
p = _____;
pitch = pitch - _____ + 1;
```

```

pr = zeros(length(_____),length(_____));
ind = 1;
for i = 1:length(pitch),
    for j = 0:duration(i)-1,
        if pitch(i) > 0,
            pr(_____,_____) = col;
        end
        ind = ind + 1;
    end
end
end
end

```

練習 2 下線部を埋めて piano_roll_matrix を完成させよ。

この関数は次のような使い方もできる。

```

>> p1 = [60 60 67 67 69 69 67];
>> d1 = [1 1 1 1 1 1 2];
>> p2 = [55 55 57 55];
>> d2 = [2 2 2 2];
>> p3 = [52 52 53 52];
>> p4 = [48 48 48 48];
>> [pr1,t,p]=piano_roll_matrix(p1,d1,48,69,1,1);
>> [pr2,t,p]=piano_roll_matrix(p2,d2,48,69,1,2);
>> [pr3,t,p]=piano_roll_matrix(p3,d2,48,69,1,3);
>> [pr4,t,p]=piano_roll_matrix(p4,d2,48,69,1,4);
>> imagesc(t,p,pr1+pr2+pr3+pr4)
>> axis xy;

```

次のような表示になる。

