

Math W84, Tue 19-Jan-2016 -- Tue 19-Jan-2016

Tuesday, January 19th 2016

day09, Tu

Topic:: Transformations

Assignment

1. Download the file `touchtone.dat` and use the FFT to determine what telephone number has been recorded in this data. Further instructions are found in the class notes of Day 8.

Where you could go from here: make a recording of yourself dialing a touch-tone phone number. How you supply such a recording to others in your group is a technical detail you would need to figure out, along with how you import one you receive into Octave. If you succeed, you can see if the detection of the numbers in the number goes exactly the same as in the above problem.

2. Download the suite of tools that includes

```
inversion.m
transpose.m
retrograde.m
playMelody.m
makeMelody.m
someTunes.m
```

You only get the melodies in `someTunes.m` by typing

```
> someTunes
```

at the Octave prompt. If you type

```
> who % whos, with an 's', also works, and gives greater detail
```

you will see all the names of things stored. There are things like `twinkle`, which is the song you would guess. Type

```
> twinkle
```

and you see how it is defined. It is entered as a matrix where each row corresponds to a note (first column) and a duration (2nd column) measured as a fraction of one beat. The note 0 corresponds to some reference note, a note you must supply as a frequency when you call `makeMelody`. A note of (-1) is one half (as per equal-temperament) below the reference note; 19 is one octave plus a fifth above the reference.

You can play the song using the two commands `makeMelody` and `playMelody`. The first of these uses your encoding to make a playable vector, one that `playMelody` will accept.

```
> ys = makeMelody(twinkle, 392, 80)
      % 392 is G4, the reference note 0
      % Tweak 80 up or down to change the tempo
> playMelody(ys)
```

Things get more interesting when you do a transformation

```
> rtwinkle = retrograde(twinkle);  
> ys = makeMelody( rtwinkle, 392, 80);  
> playMelody( ys )
```

or if you want to play twinkle together with its retrograde inversion:

```
> ritwinkle = retrograde(inversion(twinkle));  
> ys = makeMelody( ritwinkle, 392, 80);  
> ys += makeMelody( twinkle, 392, 80);  % add original melody  
> playMelody( ys )
```

Play with these songs and tools. Try out various transformations like J.S. Bach on his goldbergAria. See if you can play, and then improve upon, the mirror violin duet (the "table canon"). How does the tune musOff sound with its retrograde or its retrograde inversion?

3. I implemented `inversion.m` so that you can specify a note-center about which the inversion occurs—a possibly different zero line than the note we are calling 0. By default, the zero line corresponds to the 0 note. Try it out. See if you can find a sequence involving `inversion.m` *without* specifying a different zero line paired with calls to `transpose.m` which achieve the same thing as

```
> inversion( songname, 7 )  % inversion about the scale's 5th
```

4. A pair of transformations I did not implement are
augmentation or diminution

See if you can write an `.m` file and incorporate it into your experimentation above.

5. Take some other tune of your own choosing—a single line of music—and try out these tools on your line.
6. Read Sections 9.1–9.7 in the Benson text, and answer the accompanying questions at Socratic.com.