Math W84, Thu 21-Jan-2016 -- Thu 21-Jan-2016

---

Thursday, January 21st 2016

---

day11, Th

Topic:: math in musical composition

Topic:: 12-tone music

### Assignment for Day 11

Compose your own 12-tone piece. The general guidelines (rules to follow) are found at the website https://carolingianrealm.blog/HowToWriteA12ToneComposition.php.

No 12-by-12 grid needs to be submitted by you, just a song. A really basic song can consist of transformations of a tone row. A really boring tone row, where notes are played one at a time, all lasting one beat, might be this one:

```
> myToneRow = [1 1; 0 1; 3 1; 2 1; 5 1; 4 1; 7 1; 6 1; 9 1; 8 1; 11 1; 10 1];
```

Then you can throw together different instances of transformations of this tone row, things like

- inversion.

  ```
  > inversion(myToneRow)
  ```

  My `inversion.m`, by default, inverts about the horizontal center of the note you are calling 0. That is, 0 will stay fixed, 1 (a half step up) will be turned into (-1) (a half step down), the note labeled 5 will be inverted to (-5), etc. You can

  - make the inversion happen at a different horizaontal center. The command

    ```
    > inversion(myToneRow, 3)
    ```

    will leave the note labeled 3 fixed, will take the note called 4 to the one called 2 (and vice versa), the note called 8 to (-2) (and vice versa), etc.
  - You will notice that inversion (along with some transpositions) takes some notes with positive number designations to ones with negative number designations, effectively placing them in a differen octave. If you want to return everything to the original octave, you can use the mod command:

    ```
    > mod( inversion(myToneRow, 3), 12)
    ```

- transpositions. The command

  ```
  > tranpose(myToneRow, 3)
  ```

  transposes everything up 3 steps. to transpose down, make the second argument a negative.

- retrograde.

A song that uses both the retrograde of the tone row transposed up 2 steps, and the inversion of the tone row about tonal center 5 (but maintaining the same octave), might look like this:

```
> mySong = [retrograde(tranpose(myToneRow, 2)); mod(inversion(myToneRow, 5), 12)];
> playMelody( makeMelody( mySong ) )
```