What I do

Callum Cassidy-Nolan

Understanding, Knowledge and Construction

I enjoy entering new domains and understanding the domain from the ground up. I follow the continuous development paradigm as I learn, so I am never "done" studying or learning in a domain, I still study linear algebra, calculus, physics and other domains even when I am not taking courses on them.

I store what I know and learn in a graph database so that I can have a place where I build up my knowledge and have the ability to retrive information through complex queries like "Show me all the theorems which use the definition I just learned".

Finally I truly enjoy construction of ideas and objects that can not only be thought about, but be perceived through other senses focusing mostly on visual and auditory. My favorite types of things to construct are ones that require creativity, skill and interaction between humans, some examples would be coming up with proofs, explanations, artwork, music, and physical activities. You can read more about each one of these areas on my website.

Mathematics

My areas of expertise:

- Single and multi-variable calculus
- Probability
- Combinatorics
- Complex Numbers
- Linear & Abstract Algebra
- Group Theory
- Number Theory
- Graph Theory

A sample proof I came up with for linear algebra

A Fundamental result between Linear Transformations and Matricies

- Let $T : R^m \to R^n$ have an inverse, $\mathcal{E}$ and $\mathcal{E}'$ be bases for R^m and R^n respectively.
- Define $L = T \circ T^{-1}$, that is for any x in R^n, $L(x) = T(T^{-1}(x))$. Then the matrix which induces L is a matrix K such that:

$$[L\vec{x}]_{\mathcal{E}'} = K[\vec{x}]_{\mathcal{E}}$$

- Since $L(x) = I(x) = x$, we know that K is the identity matrix. That is

$$[L\vec{x}]_{\mathcal{E}'} = I[\vec{x}]_{\mathcal{E}'} \quad (\gamma)$$

- Take M to be the matrix which induces T_m:

$$[T(\vec{v})]_{\mathcal{E}'} = M[\vec{v}]_{\mathcal{E}}$$

so replacing $\vec{v}$ with $T^{-1}(x)$ and recalling that $L(x) = T(T^{-1}(x))$ then from the above equation, we derive:

$$[L(x)]_{\mathcal{E}'} = [T(T^{-1}(x))]_{\mathcal{E}'} = M[T^{-1}(x)]_{\mathcal{E}} \quad (\alpha)$$

- Supposing that A is the matrix which induces T^{-1}

$$[T^{-1}(\vec{v})]_{\mathcal{E}} = A[\vec{v}]_{\mathcal{E}'}$$

then we can replace the right hand side of α with $MA[\vec{x}]_{\mathcal{E}'}$, in tandem with γ that is:

$$I[\vec{x}]_{\mathcal{E}'} = [L\vec{x}]_{\mathcal{E}'} = MA[\vec{x}]_{\mathcal{E}'}$$

Therefore $MA = I$, to get $AM = I$, repeat the proof for $G(x) = T^{-1}(T(x))$. Thus We may conclude that for any linear transformation T, the matrix which induces it is the inverse of the matrix which induces T^{-1}.

Teaching

Anyone can understand any concept as long as the follow a bottom up approach. By knowing all the definitions and theorems that are in involved in the current piece of knowledge you are trying to obtain, then you have set yourself up for success, now you can freely use your skills without lacking any information. I have been a member of the Math Learning Center at U of T and heavily used this technique with students and seen it work well. If you are interested in learning about any of the concepts on this page, then I can tutor you in it (email me).

Programming

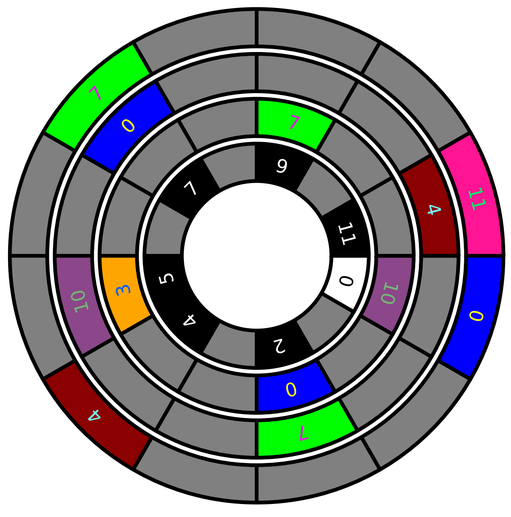
- Skills
  - Creativity - I try to implement my own version before looking at a potential solution - I also enjoy visualizing algorithms.
  - Effiency - I put time in to use my tools as efficiently as possible.
  - Debugging - GDB, print statements, and going line by line through my code to spot them.
  - Usability - I take time to make sure my programs can be used in an efficient and simple manner
  - Documentation - I write documentation and comments explaining complicated code segments for other programmers and my future self.
  - Not afraid of reading manuals and documentation - Some of my favorite pages are from:

    `:tab help user-manual`

  - Know where to get help:
    - IRC, stack exchange, real life programmers
- Specifics
  - Python, C/C++, Shell Scripting
  - Unix Utilities
    - Vim, Tmux, grep, sed, ssh, ...
  - Linux
    - An understanding of how each of the components of a linux machine come together: Kernel, X server, processes, init systems, grub, ...
    - I can operate just fine on a machine that has no graphical interface.
  - Network & Socket Programming
  - Graphics Programming
    - OpenGL, Processing, PyGame
  - Game Making
    - Physics Engines (frame independent gameplay)
    - ENet Library for multiplayer

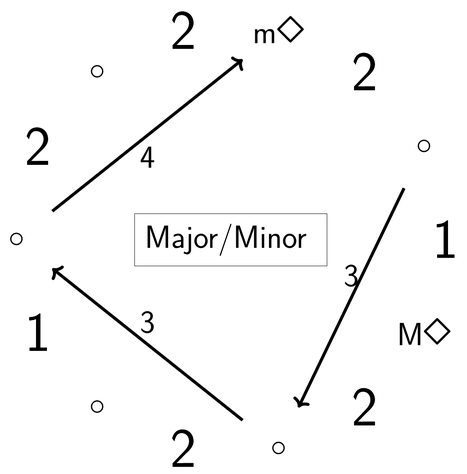
Output from a C program I desgined using the cairo drawing library:

A 2 5 1 or (2 7 0) chord progression moving radially outward.



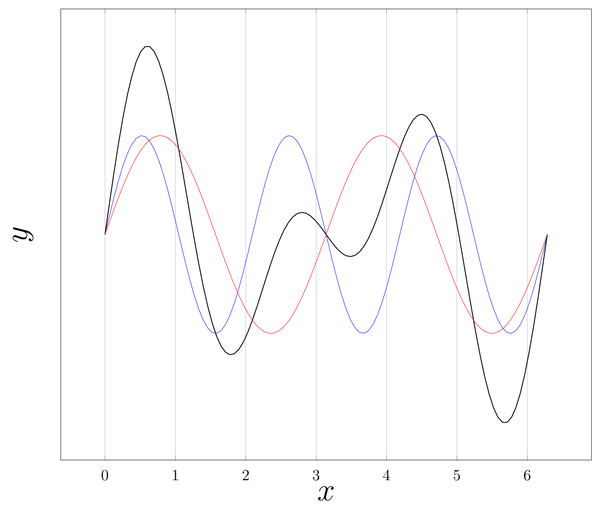
cuppajoeman

Music



I have developed an equivalent representation of the standard notation and concepts that have been derived in standard music theory that allows results to be derived in a much simpler fashion. For example, the circular sequence you see in the above diagram allows us to derive and understand where the diatonic chords come from, the annotated chord would be 0, 3, 6, 10 which is a half-diminished 7th chord.
The equal temperment system approximates just intonation, I am interested when multiple tones are played together and how we can understand why some of these intervals sound more consontant or dissonant, below is the representation of the sound produced by a perfect 5th, the graph represents the sum of two waves, one wave oscillates twice while in that same time the other oscillates thrice, a ratio of $2 : 3$

$$f(x) = \sin(3x) + \sin(2x)$$



I believe there is a connection between simplicity and beauty, and that after the unison/octave, the perfect 5th is the next most consontant interval, because it uses the next most simple ratio. For two tones played together and their ratio $r_1 : r_2$ I define their simplicity by $\operatorname{lcm}(r_1, r_2)$, ordering by their simplicity we get: (Do you agree?)

unison, perfect 5th, perfect 4th, Major 6th, Major 3rd, Minor 6th, Minor 7th, Major 2nd, Major 7th, Minor 2nd, Tritone

Semantic Data

- After using physical notes, then sorting directories in a unix filesystem for a long time, I started using a graph database to store my information. I queried it directly, this worked well but I wanted others to be able to add to the graph too but the interface I had built only suited my needs.
- To allow others to contribute I started a wiki using the software "Mediawiki", using the "Semantic Mediawiki" plugin where I learned about the semantic web. Here I gained the ability for others to contribute, but lost the sense of the graph which I tried hard to model.
- My final approach was to make the system more modular by using the best service for each part of my knowledge LaTeXfor formatting, git for version control and a graph database to query the data, to pull this off I separated the data from the structure. This is my current setup and you can view and contribute to the graph here.

  - You can view images of each of the above iterations here.