

B3B: Functional Sound Processing

CS1101S: Programming Methodology

Martin Henz

August 27, 2021

- 1 What is sound?
- 2 Digital Sound
- 3 Digital music

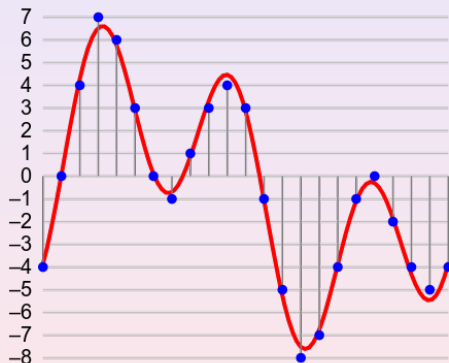
- 1 What is sound?
- 2 Digital Sound
- 3 Digital music

What is sound?

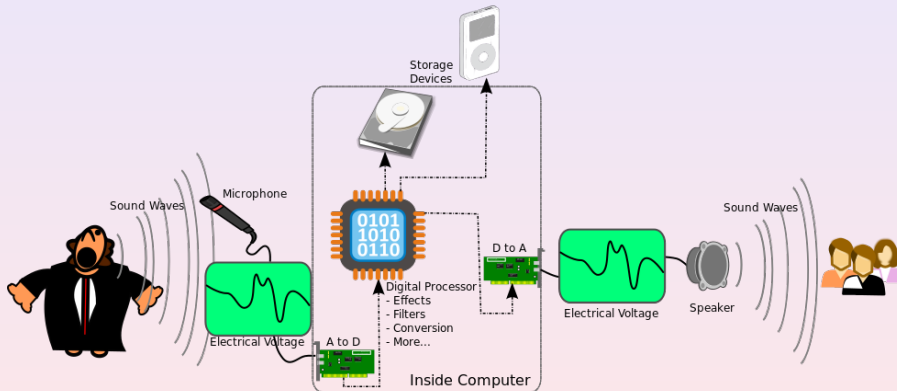
What is sound?

Watch spherical pressure waves

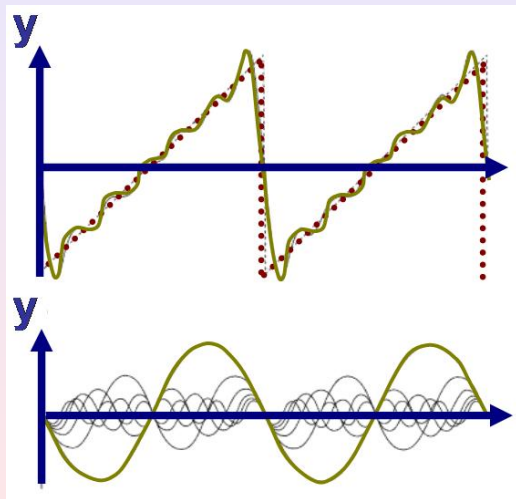
Digital Sound



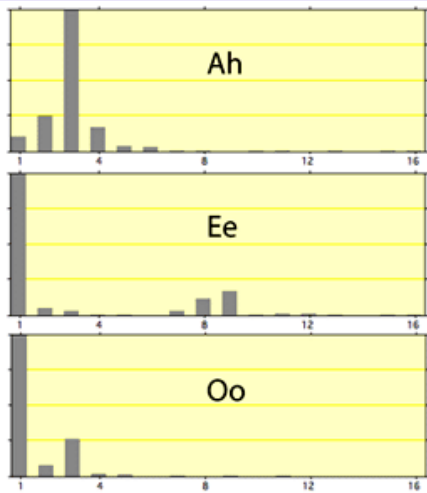
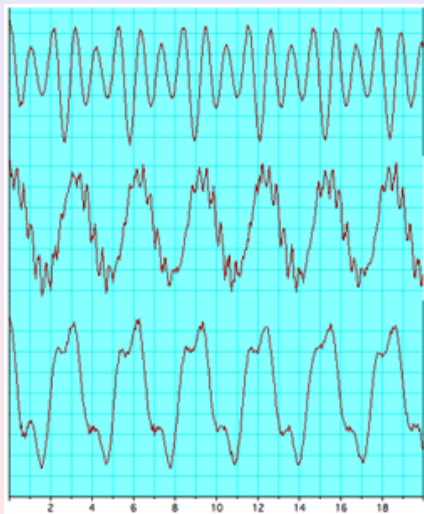
Digital Sound Processing



Fourier Analysis



Analysing sound in the frequency domain



Pre-history: Analog Electronic Synthesizers

<https://www.youtube.com/watch?v=1cew7dAbDh0>

go to 2:29

Fast forward: Today's tools for making music

Tone matrix

Everyone is a composer!

<https://www.maxlaumeister.com/tonematrix/>

Techno 2021

<https://www.youtube.com/watch?v=gshky0hLNS8>

Live coding

Live coding

What is it?

Creating sound and images using programming during a live performance on stage

Live coding

What is it?

Creating sound and images using programming during a live performance on stage

- Some DJs do “live coding”, but typically without showing the programs to the audience

Live coding

What is it?

Creating sound and images using programming during a live performance on stage

- Some DJs do “live coding”, but typically without showing the programs to the audience
- Domain-specific languages (DSL) for live coding

Live coding

What is it?

Creating sound and images using programming during a live performance on stage

- Some DJs do “live coding”, but typically without showing the programs to the audience
- Domain-specific languages (DSL) for live coding
- Andrew Sorensen, Five over Four, 2017, using a Scheme-based DSL <https://www.youtube.com/watch?v=FYWt1V4JV6o>

Functional Sound Synthesis

Simple example:

```
const pitch_A_wave =  
    t => math.sin(2 * math.PI * 440 * t);  
const pitch_A =  
    make_sound(pitch_A_wave, 1.5);  
play(pitch_A);
```

Functional Sound Synthesis

Complex example:

```
const pitch_A_wave =  
    t => math_sin(2 * math_PI * 440 * t);  
// many more functions, all continuous  
const final_sound = ...;  
  
play(final_sound); // digitize and make audible
```

Functional Sound Synthesis

Complex example:

```
const pitch_A_wave =  
    t => math.sin(2 * math.PI * 440 * t);  
// many more functions, all continuous  
const final_sound = ...;  
  
play(final_sound); // digitize and make audible
```

In our missions, *synthesise* (compose) sounds as functions in the continuous domain, and then digitize the result.

Summary: Functional Sound Synthesis

Summary: Functional Sound Synthesis

- Sound is a physical phenomenon

Summary: Functional Sound Synthesis

- Sound is a physical phenomenon
- We digitize sound to process it in the computer

Summary: Functional Sound Synthesis

- Sound is a physical phenomenon
- We digitize sound to process it in the computer
- To hear digitized sound, we need to “undigitize” it

Summary: Functional Sound Synthesis

- Sound is a physical phenomenon
- We digitize sound to process it in the computer
- To hear digitized sound, we need to “undigitize” it
- In CS1101S, we do it differently: we work with *functional* sound