# computers and music (1)

representing music digitally

## learning goals:

- knowledge of (some) musical activities that can be supported by computers
- familiarity with the basic method for encoding a sound signal digitally, including the guideline for sampling frequency
- knowledge of different levels of music representation and their uses, in particular acoustic and score level
- basic familiarity with musical parameters and their representation in GUIDO Music Notation

## Lady Ada's Vision

*[The Analytical Engine] might act upon other things besides number, [. . . ] the engine might compose elaborate and scientific pieces of music of any degree of complexity or extent.*

Augusta Ada King, Countess of Lovelace (1815–1852)

## Working With Music

**(Some) musical activities / processes:**

- creating (composition, improvisation)
- playing (performance)
- listening (perception)

**but also:**

- building instruments
- analysing
- notating / copying / publishing / distributing

(even analysing and designing accoustics of instruments and concert halls)

⇝ computers can support all these activities

**Musical objects / data:**

- instruments
- scores (written notes, rests, other symbols)
- recordings

**as well as less tangible "objects", such as:**

- works (symphonies, songs, sonatas, . . . )
- melodies, rhythms, voices
- interpretations, performances

⇝ need appropriate representations to work with them algorithmically

*Any language begins as music and ends up being an algebra.*

André-Marie Ampère,
French mathematician and physicist (1775–1836)

## Music Representation

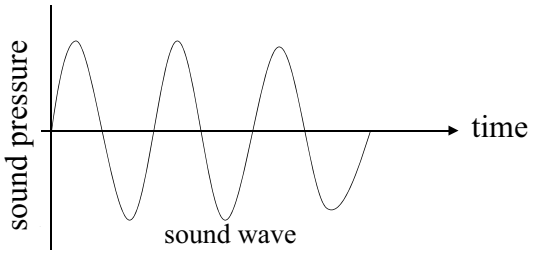
To work with music algorithmically,
we need a formal music representation.

But what should we represent?

- Sound? (like on a record, tape, or CD?)
- Score? (like in sheet music)
- Structure? (musical form, "architecture")
- Properties? (statistical features, number of notes, *etc.*)

And how?

## sound waves

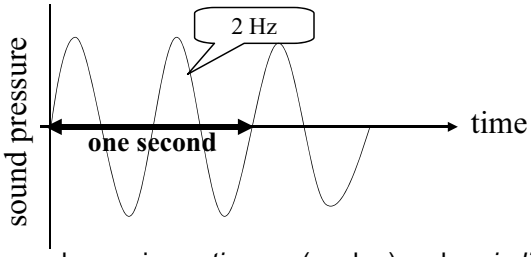



1. source of sound vibrates

2. pressure waves (vibrations) emanate from the source

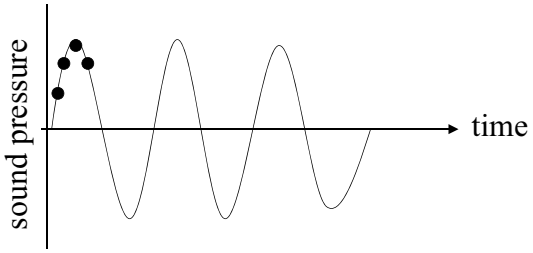
3. waves are sensed by nerves in ear

## properties of sound waves



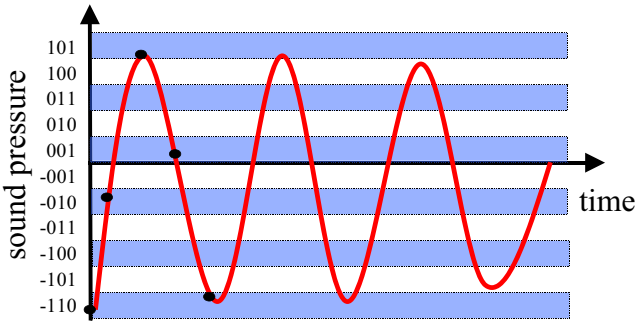

- a sound wave is *continuous* (analog) and *periodic*
- the *frequency* of a sound wave, measured in Hertz (Hz), is the number of periods per second
- audible sound waves have frequencies ranging between 20Hz and 20 000 Hz (20 kHz)

## digitising sound



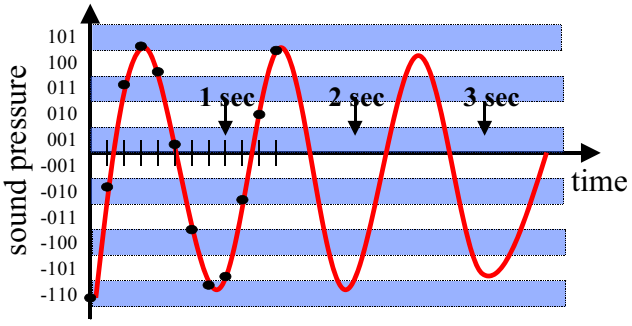

- *sampling* is used to digitize analog sound waves
- a ***sample*** is a measurement of the pressure at a point in time

## samples are measured in *bits*




- sample sequence: -110, -010, 101, 001, -110, …
- in this example, 4 bits per sample (1 for the +/- sign)

## sampling rate: # samples per second



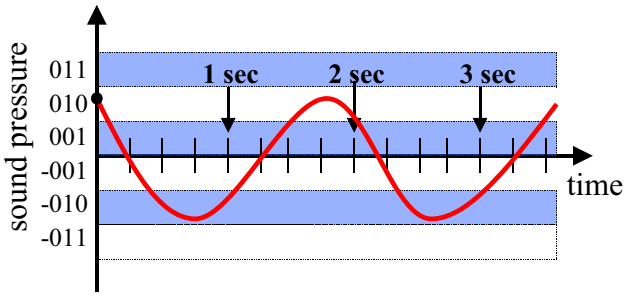

- sampling *rate* here is 8 samples/sec

# digitised sound quality

- for a given sound wave, the quality of the digitised sound data is determined by
  - the *sampling rate*
  - the *number of bits* per sample
- **guideline:** for good quality, the sampling rate should be at least twice the maximal frequency

  *Example:* Want to capture frequencies up to 20 000 Hz => need a sampling rate of at least 40 000 samples/sec

# exercise




What is the digitised sound representation for this sound wave, when the sampling rate is 4 and the number of bits per sample is 4?

# exercise

A sound wave that extends for 1 minute is sampled at a rate of 44 000 samples per second. Each sample is 16 bits.

What is the total number of bits needed to represent the sample sequence for that sound wave?

# advantages of digitised sound

- can be copied without loss of quality
- can be edited in complex ways, *e.g.,*
  - remove coughing from live recording
  - speed up or slow down the speed of the music without changes in pitch
- can be compressed to save storage space, *e.g.,* by removing non-audible aspects of sounds (this provides the basis for MP3 coding)

**Levels of music representation:**

- *acoustic / physical (level 0)*
  music as sound, represented as waveform
  (*e.g.*, CD recording, WAV file, MP3 file, ...)
  *This is what instruments produce and what we hear.*
- *score / notation (level 1)*
  explicitly represents musical parameters,
  such as pitch, duration, loudness, instrument
  *This is (mostly) what composers write and musicians play from.*

**Levels of music representation (continued):**

- *structure (level 2)*
  explicitly represents musical structure,
  including movements, repeats, recurring material
  *Example:* *A C B C A C*, where *A*, *B*, *C* are pieces of music,
  *e.g.*, the verses and chorus of a song.
- *metastructure (level 3)*
  explicitly represents algorithms for composition and analysis,
  automata, grammars, functions, generators
  *Example:* A program that uses a random number generator to create musical fragments and combines them into a piece.

Sound recording, editing, mixing, playback typically happen at the acoustic level (=level 0).

$\rightsquigarrow$ Musical parameters not represented explicitly (very difficult to access and manipulate).

*But:* Many creative musical activities (including most forms of composition) use musical parameters explicitly.

$\rightsquigarrow$ Need score (= level 1) or higher level representation.

**Conventional music notation**

- complex, expressive graphical language
- historically evolved system, originated ca. AD 1000
- optimised for performance, but equally used for composition, analysis, ...

**Elements of music notation: Primary musical parameters**

- pitch (related to physical frequency) specified by
  - pitch class (note name):

    c | c-sharp = d-flat | d | d-sharp = e-flat | e | f | f-sharp = g-flat | g | g-sharp = a-flat | a | a-sharp = b-flat | b

  - register (octave number):
    $-3,\ -2,\ -1,\ 0,\ 1,\ 2,\ 3,\ \ldots\ 8$

  *pitch classes repeat in each register:*
  c1 d1 e1 f1 g1 a1 b1 c2 d2 ...

**Elements of music notation: Primary musical parameters (continued)**

- note values (related to physical duration)
  specified as fractions: $1/1,\ 1/2,\ 1/4,\ 1/8,\ 1/16,\ \ldots$

**Additional elements of music notation: Secondary musical parameters**

- tempo (speed)
- intensity (loudness)
- timbre (instrument / style of playing)

**GUIDO Music Notation [Hoos et al., 1996–2001]**

- represents music notation (level 1)
- plain text, human-readable
- *representationally adequate*:
  simple things have simple representations

**GUIDO Music Notation: Primary musical parameters**

- pitch classes:

  c | c-sharp = d-flat | d | d-sharp = e-flat | e | f | f-sharp = g-flat | g | g-sharp = a-flat | a | a-sharp = b-flat | b

  `c c#/d& d d#/e& e f f#/g& g g#/a& a a#/b& b`

- pitch: ⟨*pitch class*⟩ ⟨*register*⟩
  *e.g.*: `c1 a2 f#-1 a-3 c5`
- durations: `/1, /2, /4, /8, ...`

**GUIDO Music Notation: Notes and Rests**

- Notes: ⟨*pitch*⟩⟨*duration*⟩
  *e.g.*: `c1/4  a2/8  f#-1/1  b&/16`
- Rests: `_`⟨*duration*⟩
  *e.g.*: `_/4  _/2  _/8`

**GUIDO Music Notation: Sequences and Segments**

**Sequence:**
Series of notes (or rests) that are played sequentially (one after the other)

In GUIDO: `[`⟨*note/rest*⟩ ⟨*note/rest*⟩ . . . ⟨*note/rest*⟩`]`

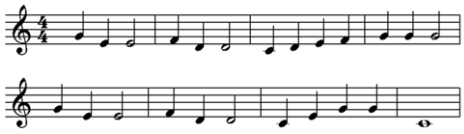
*E.g.:* `[c1/4 d1/4 e1/4 f1/4 g/2]`

**Segment:**
A set of sequences (voices) that are played concurrently (at the same time)

In GUIDO: { ⟨*sequence*⟩ , ⟨*sequence*⟩ , . . . , ⟨*sequence*⟩}

*E.g.:* {`[c1/4 d1/4 e1/2],`
`[e1/4 f1/4 g1/2]`}

**GUIDO Music Notation (simple example)**

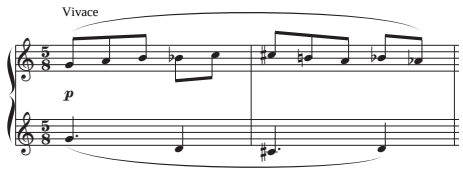


```
[ g1/4 e1/4 e1/2  f1/4 d1/4 d1/2  c1/4 d1/4 e1/4 f1/4  g1/4 g1/4 g1/2
  g1/4 e1/4 e1/2  f1/4 d1/4 d1/2  c1/4 e1/4 g1/4 g1/4  c1/1 ]
```

```
[ c1/4 e1/4 g1/8 f1/8 e1/8 d1/8 c1/2 ]
```

```
[ c1/16 c# d d# e f f# g g# a a# b c2/4 ]
```

**GUIDO Music Notation (slightly more complex example)**




```
{ [ \meter<"5/8"> \tempo<"Vivace"> \intens<"p">
    \slur(\beam(g1/8 a b) \beam(b& c2) | \beam(c# b1 a b& a&)) ],
  [ \slur(g1/4. d/4 | c#/4. d/4) ] }
```