# CONVOLUTION

# Mathematik

$$a[n] * b[n] = output[k] = \sum_{n=0}^{N-1} a[n] \times b[k-n]$$

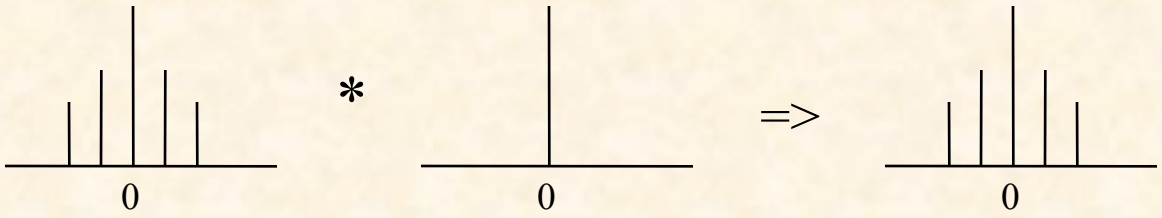
length(output) = length(a)+length(b) - 1

# Direct Convolution

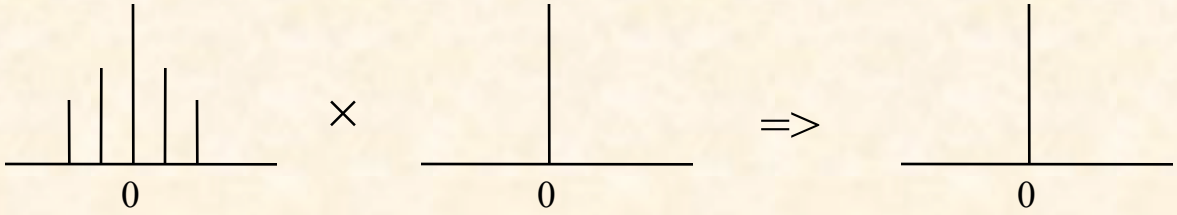
a = [1 0.75 0.5 1] Impulse Response b = [0.1 1 0 0.5]

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| c[1] | 1*0.1 | 0.75*0.1 | 0.5*0.1 | 1*0.1 | | | |
| | | + | + | + | | | |
| c[2] | | 1*1 | 0.75*1 | 0.5*1 | 1*1 | | |
| | | | + | + | + | | |
| c[3] | | | 1*0 | 0.75*0 | 0.5*0 | 1*0 | |
| | | | | + | + | + | |
| c[4] | | | | 1*0.5 | 0.75*0.5 | 0.5*0.5 | 1*0.5 |
| | = | = | = | = | = | = | = |
| result | 0.1 | 1.075 | 0.8 | 1.1 | 1.375 | 1.25 | 0.5 |

Convolution:




Multiplikation:

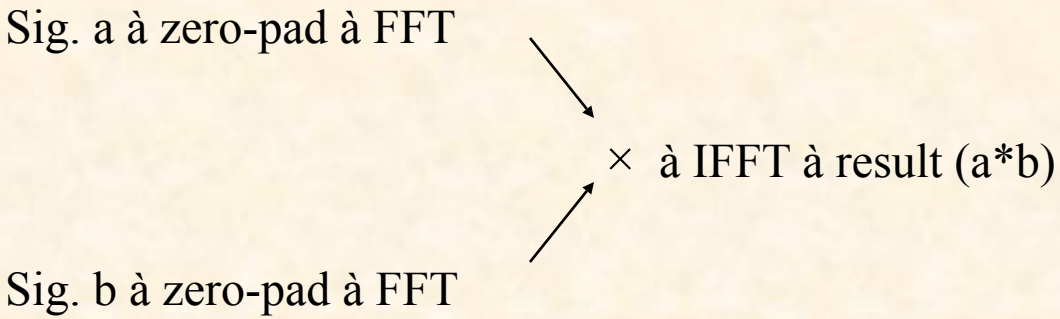



*Convolution ist Multiplikation im Frequenzbereich!*

# Fast Convolution:

Produkt von zwei N großen DFT‘s ist gleich der Convolution von zwei N großen Signalen:

⇒Convolution durch zero-padding, FFT‘s der Eingangssignale, Multiplikation und anschließender IFFT.



Convolution ⟷ digitale Filter

FIR-Filter:

$$y[n] = ( x_1 \times a[n] ) \pm ( x_2 \times a[n-1] ) \pm \ldots ( x_i \times a[n-j] )$$

-

$$y[n] = \sum_{m=0}^{N-1} x[m] \times a[n-m]$$

è *Jeder FIR-Filter kann als Convolution bezeichnet werden*.

# Convolution in C-Sound

## Opcode „convolve“:

ar1[,...[,ar4]]]  convolve  ain,ifilcod [, ichannel]

- Fast Convolution

- C-Sound Utility „cvanal“ konvertiert Soundfile in .cv Datei
  à *-b (begin), -d (duration), -c (channel), -s (sample rate)*

## Opcode „dconv“

ar  dconv  asig, isize, irfn

- direct Convolution
- Keine Voranalyse nötig

# Faltungshall:

```
instr 1
iamp = 0.2

ain1 diskin "pattern2.wav", 1
a1 convolve ain1, "smallchamber.cv",1

out a1*iamp

endin
```

```
instr 1
iamp = 0.2

ain1 diskin "pattern2.wav", 1
a1 convolve ain1, „bigchurch.cv",1

out a1*iamp

endin
```

```
instr 2


iamp = .005


ain1 diskin "pattern2.wav", 1
a1 convolve ain1, "pattern2.cv“ ; erste 2 sek.

imix = p4

adel   delay    imix*ain1,  2

out a1*iamp+adel

endin

;Score:
i2 0 17   .1
i2 +  .     .9
i2 +  .     .3
e
```

```
instr 3

iamp   =  .008
itable =  1
isize  =  ftlen(itable)

ain diskin "sheila.aiff", 1

a1 dconv ain, isize , itable

out a1*iamp

endin

;score
f1 0 65536 1 "bell.aiff" 0 0 1
i3 0 10
e
```