

Joachim Heintz

**Views on Sine Waves**

Online Workshop at TIEMF 2021

Part II:

Arrays and Spectra

Please **download the example files now** at:  
[www.joachimheintz.de/workshops/tehran\\_2021](http://www.joachimheintz.de/workshops/tehran_2021)  
-> examples\_II.zip

Unpack the examples\_II.zip in your working directory.  
(I suggest to create a new folder.)

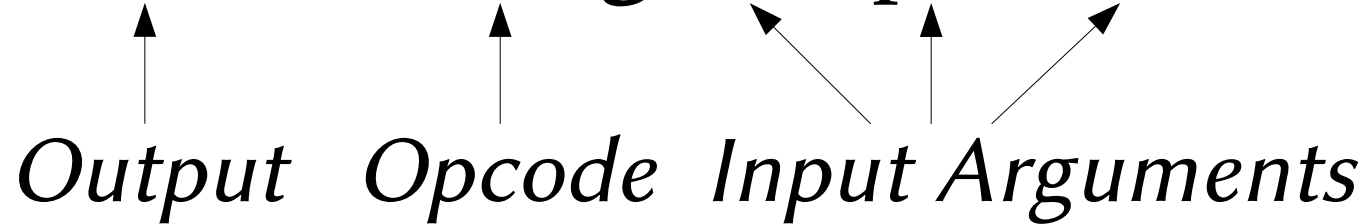
# **1 Coding in Functional Style**

(Csound Advanced ...)

## Traditional Csound Code

`aEnv linseg 1, p3, 0`

*Output Opcode Input Arguments*



## Functional Style

`aEnv = linseg:a(1, p3, 0)`

*Output = Opcode:Rate Input Arguments*



# Limitation

Only possible for opcode with **none** ...

```
out(aPartial, aPartial)
```

... or **one** output ...

```
kRnd = randi:k(1/6, 2, 2)
```

... but not with two or more outputs:

```
aL, aR = pan2(aOut * iAmpFac, iPan)
```

# Strong Recommended

**Always** specify the rate:

```
kRnd = randi:k(1/6, 2, 2)
```

```
aEnv = transeg:a(1, p3, -4, 0)
```

```
iFreq = mtof:i(iMidiPitch)
```

Read more:

[Csound FLOSS Manual Chapter 03 I](#)

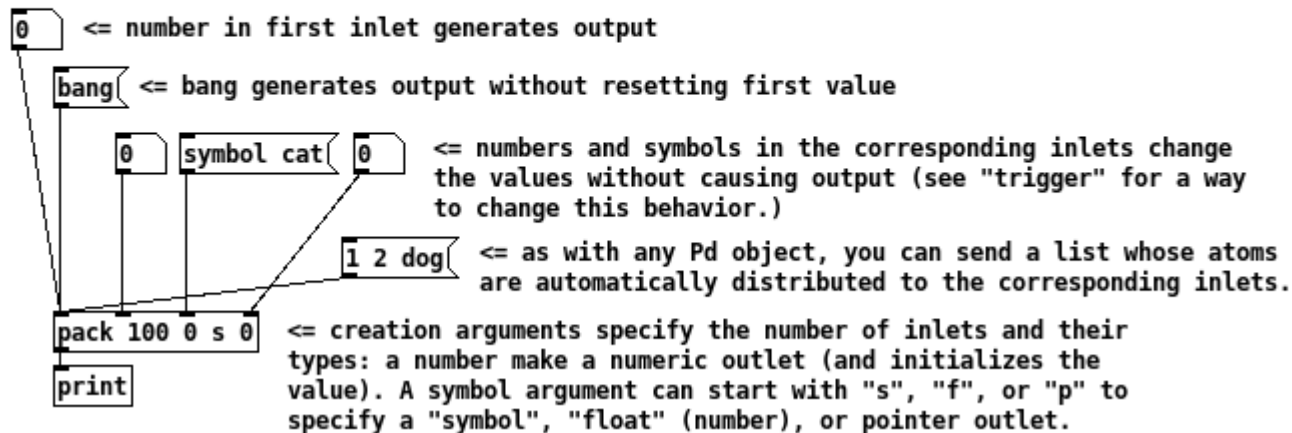
## **2 Introduction to arrays**

# Array = Collection of data

## PD and Max: List

`pack` - combine several atoms into one message

The pack object takes a series of inputs and outputs a concatenated list. The number of creation arguments determines the number of inlets.



## SC, Python and others: List and/or Array

```
mylist = [1, 2, 3, 4, 5]
```



# Array data and index

DATA	1.996	2.391	3.051	4.018	5.281	6.015	8.274
INDEX	0	1	2	3	4	5	6

Get the fourth element (= Index 3!):

```
Element = Array[3]  
-> 4.018
```

# 3 How to work with arrays in Csound

*l\_01\_fs.csd* — creating an array and retrieving an element from the array

*l\_02\_fs.csd* — using a *while* loop to get access to all elements of the array

Read more about Arrays in Csound:

Csound FLOSS Manual Chapter 03 E

## 4 Call another instrument in the *while* loop

*l\_03\_fs.csd* — general setup for one array

*l\_04\_fs.csd* — setup for two arrays of same length

## **5 Call a partial for each element of an array in the *while* loop**

*l\_05\_fs.csd* — simple instrument for partials

*l\_06\_fs.csd* — inserting more complex instrument  
(which was result of first workshop part)

## **6 Change input to proportions instead of frequencies**

*l\_07\_fs.csd* — basic approach

*l\_08\_fs.csd* — base frequency as MIDI not number

# 7 Volume and p-fields

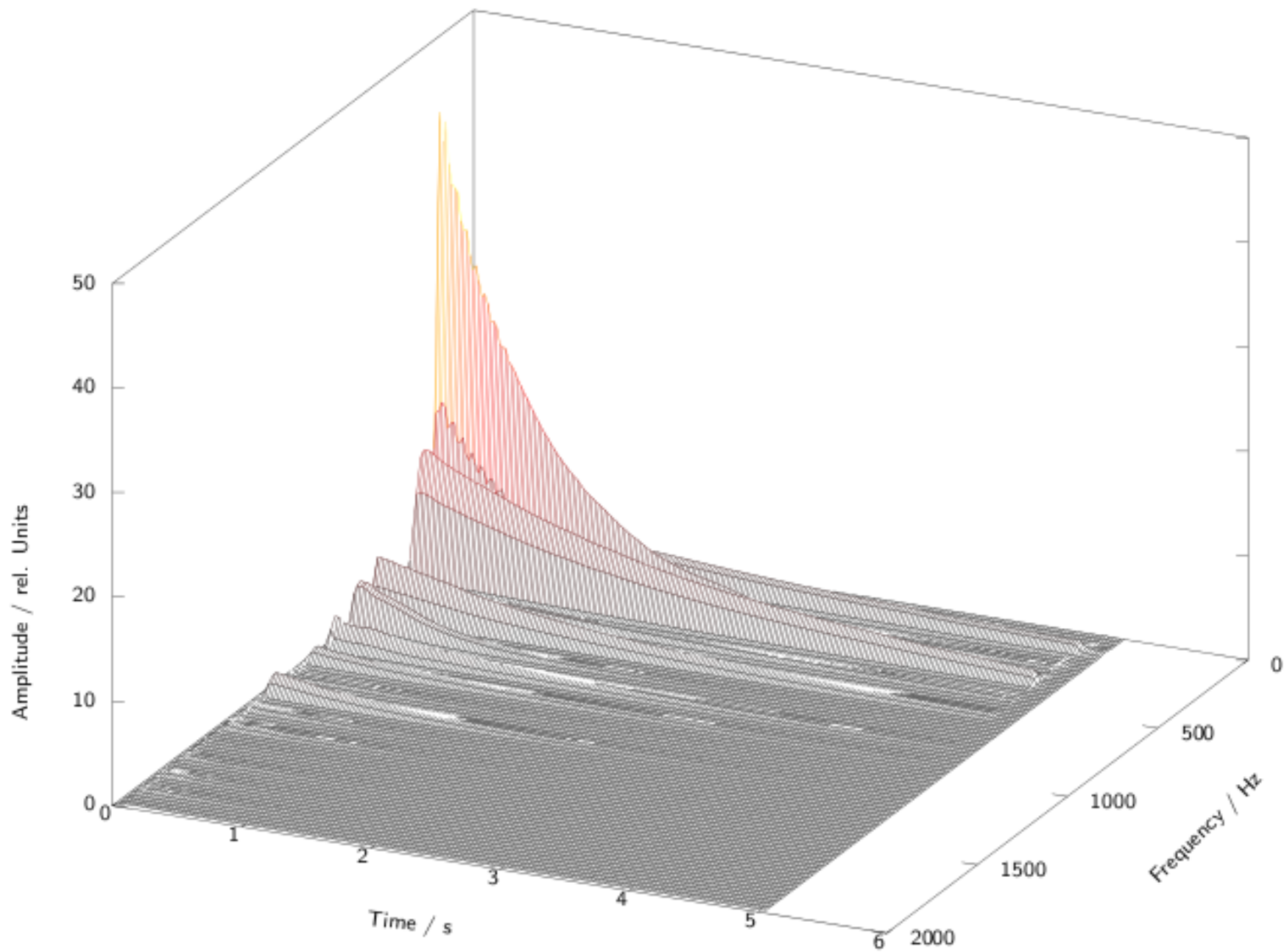
*l\_09\_fs.csd* — volume input as dB

*l\_10\_fs.csd* — volume and pitch as p-fields

## **8 Different durations for the partials (higher partials = shorter durations)**

-> figure on next page, then

*l\_11\_fs.csd* — simple implementation for it



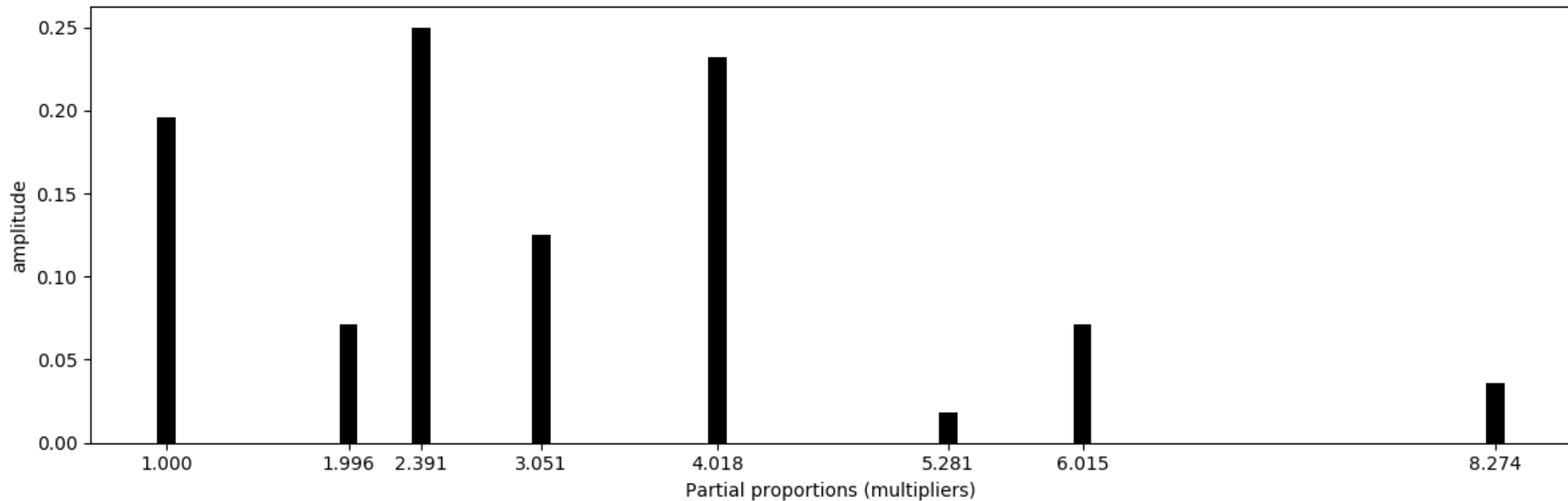


# 9 Using spectra from samples

have a look at the bell spectrum below, then

*l\_12\_fs.csd* – wine glass

*l\_13\_fs.csd* – tibetan bowl



# 10 Back to the Cloud instrument

*l\_14\_fs.csd* — cloud with wine glass