April: normales Vorlesungsgeschehen
- Lernen von Python / Signalverarbeitung anhand Kapitel 2

Anfang Mai: Bekanntgabe der Hausaufgabenthemen
jede bekommt 1 Python Skript zugeteilt

→ Clean Code Regeln anwenden
- große Prozeduren in kleine logische Abschnitte zerlegen
- große Klassen in kleine logische Abschnitte zerlegen
- assert einfügen
- DRY Don't repeat yourself
- Kohäsion erhöhen (Variablen der Klasse werden in möglichst viele Prozeduren der Klasse verwendet)

Rest vom Semester: - Vorlesungen als Spezialfach
- eine(r) stellt aktuellen Planungsstand vor und wir besprechen das gemeinsam

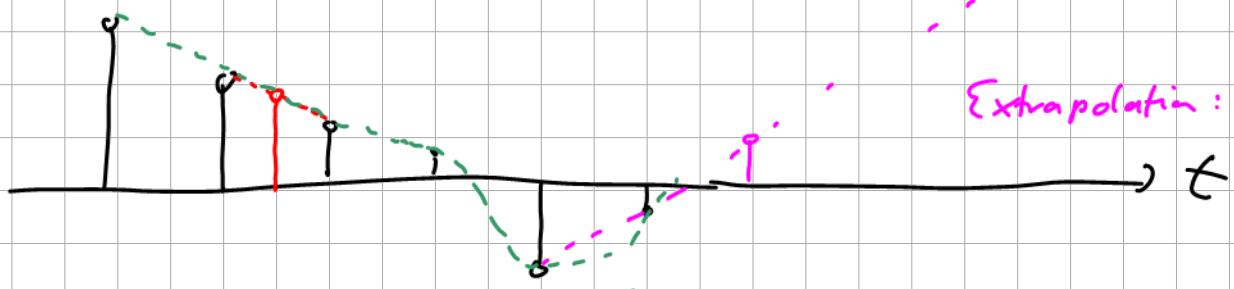
Spätestens
Am Prüfungstag: Abgabe des Pythonskriptes bis 23:59 Uhr

„Hausarbeit" ist das Pythonskript mit entsprechender Menge an Kommentaren.

Nach der Abgabe: persönliches Gespräch
ca ½ Stunde über die eigene Arbeit

→ geht in die Note ein
→ stellt den Urheber der Hausarbeit sicher.

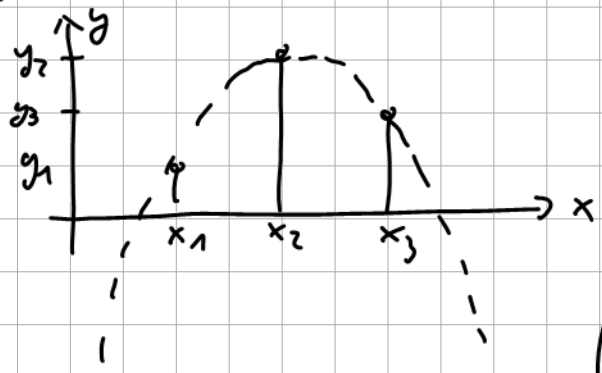
Interpolation: Stützstellen „links" & „rechts"



Extrapolation: Stützstellen fehlen

Interpolation ≙ neue Abtastung der Hüllkurve

Beispiel Extrapolation:

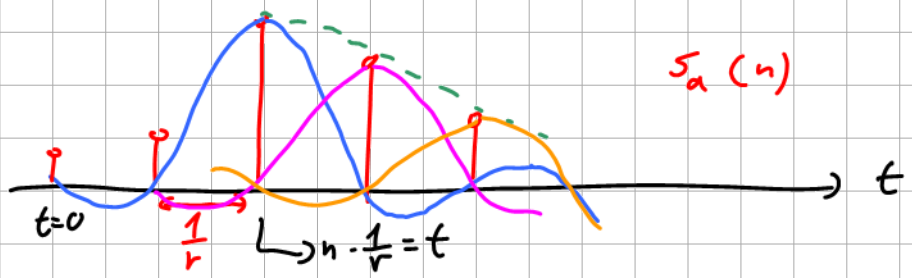


$$y_1 = a x_1^2 + b x_1 + c$$
$$y_2 = a x_2^2 + b x_2 + c$$
$$y_3 = a x_3^2 + b x_3 + c$$

$a, b, c = ?$

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} x_1^2 & x_1 & 1 \\ x_2^2 & x_2 & 1 \\ x_3^2 & x_3 & 1 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \end{pmatrix}$$

Hüllkurve ist Summe der si-Funktion.



$si(\pi t) = sinc(f)$

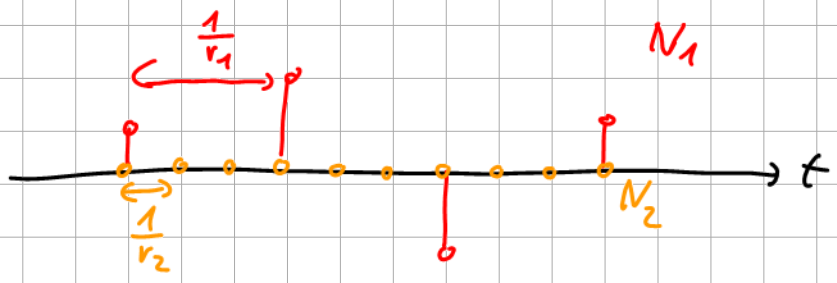
sinc Fkt mit Nullstellen bei $\frac{1}{r}, \frac{2}{r}, \frac{3}{r} \ldots$

$sinc(r \cdot t)$

sinc Fkt verschieben $t \to t - \frac{n}{r}$

$$\text{sinc}\left(r \cdot \left(t - \frac{n}{r}\right)\right) = \text{sinc}\left(r \cdot t - n\right)$$

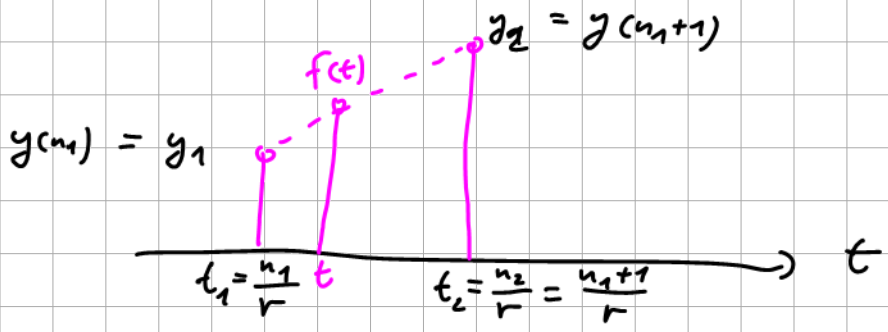
Without extrapolation upsampligfacte 3



$t_1 = \text{np.arange}(N_1) / r_1$

$t_2 = \text{np.arange}(N_2) / r_2$

lineare Interpolation



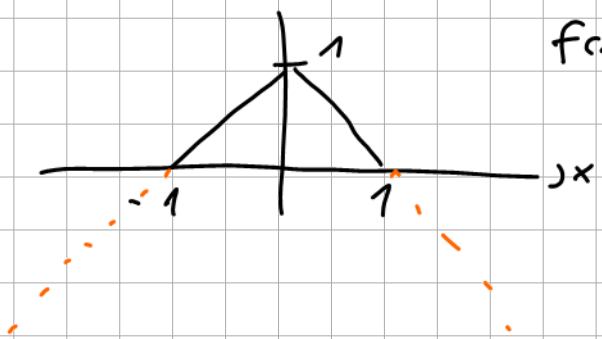
Steigung

$$m = \frac{y(n_1+1) - y(n_1)}{\frac{n_1+1}{r} - \frac{n_1}{r}}$$

$$= \frac{r \cdot \left(y(n_1+1) - y(n_1)\right)}{n_1 + 1 - n_1}$$

$$= r \cdot \left(y(n_1+1) - y(n_1)\right)$$

$$f(t) = m \cdot \left(t - \frac{n_1}{r}\right) + y(n_1)$$



$$f(x) = \begin{cases} 1 - |x| & , \text{ für } |x| \leq 1 \\ 0 \end{cases}$$

```
def Triangle(x):
    return np.maximum(1 - np.abs(x), 0.0)
```