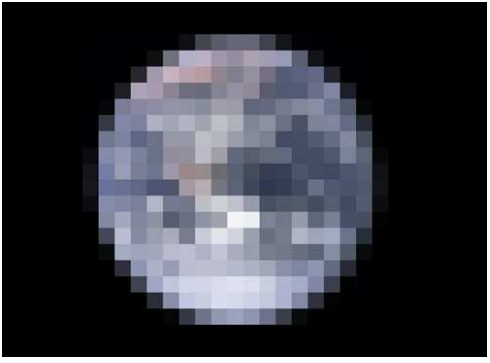
COM-202: Signal Processing

Chapter 7.a: Interpolation

overview

- the analog worldview
- bandlimited functions
- interpolation of discrete-time signals
  - Polynomial interpolation
  - Local interpolation
  - Sinc interpolation

Two models of the world



Analog/continuous versus discrete/digital

Two models of the world

digital worldview:

- arithmetic
- combinatorics
- computer science
- DSP

analog worldview:

- calculus
- distributions
- system theory
- electronics

Two models of the world, two languages

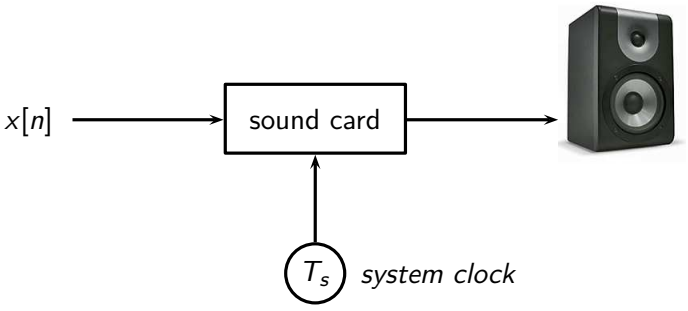
digital worldview:

- countable integer index n
- finite-energy sequences $\mathbf{x} \in \ell_2(\mathbb{Z})$
- frequency $\omega \in [-\pi, \pi]$
- DTFT: $\ell_2(\mathbb{Z}) \mapsto L_2([-\pi, \pi])$

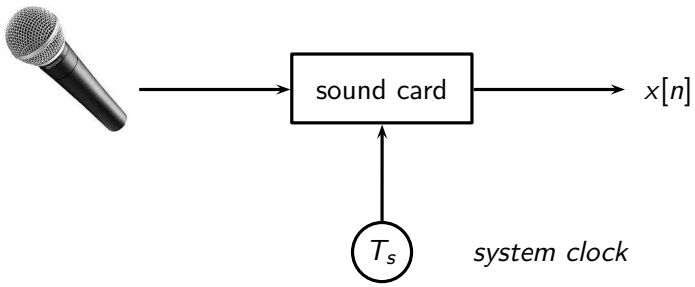
analog worldview:

- real-valued time t (sec)
- finite-energy functions $\mathbf{x} \in L_2(\mathbb{R})$
- frequency $f \in \mathbb{R}$ (Hz)
- FT: $L_2(\mathbb{R}) \mapsto L_2(\mathbb{R})$

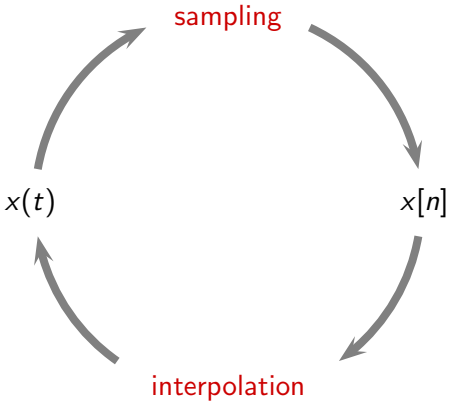
Translating between languages: interpolation



Translating between languages: sampling



Bridging the gap

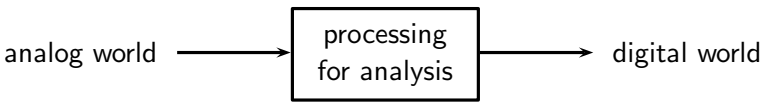


Today, processing is as digital as possible

- analog to digital
- digital to analog
- analog to digital to analog

Digital processing of signals from the analog world

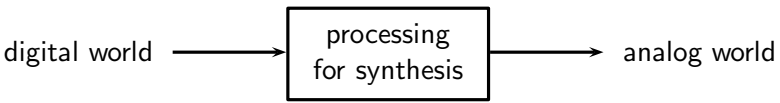
- input is continuous-time: $x(t)$
- output is discrete-time: $y[n]$
- processing is on sequences: $x[n], y[n]$



examples: storage and compression (MP3, JPG), control systems, monitoring

Digital processing of signals to the analog world

- input is discrete-time: $x[n]$
- output is continuous-time: $y(t)$
- processing is on sequences: $x[n], y[n]$



examples: telecommunication front-ends, music synthesizers, biomedical

Digital processing of signals from/to the analog world

- input is continuous-time: $x(t)$
- output is continuous-time: $y(t)$
- processing is on sequences: $x[n], y[n]$

examples: end-to-end telecommunication, sound effects, digital photography

continuous-time signal processing

About continuous time

- time: real variable t
- signal $x : \mathbb{R} \mapsto \mathbb{C}$: complex functions of a real variable
- finite energy: $\mathbf{x} \in L_2(\mathbb{R})$ (square integrable functions)
- inner product in $L_2(\mathbb{R})$

$$\langle \mathbf{x}, \mathbf{y} \rangle = \int_{-\infty}^{\infty} x^*(t) y(t) dt$$

- energy: $\|\mathbf{x}\|^2 = \langle \mathbf{x}, \mathbf{x} \rangle$

Continuous-time signal operators

main differences wrt discrete time:

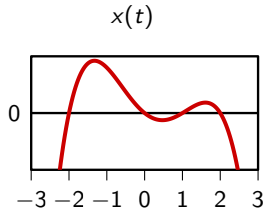
- the time shift operator works with any real-valued shift:

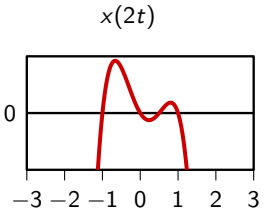
$$\mathbf{y} = \mathcal{S}^{\tau}\mathbf{x}, \quad \tau \in \mathbb{R} \quad \Rightarrow \quad y(t) = x(t+\tau)$$

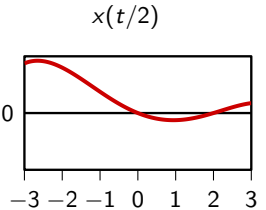
- signals can always be time-scaled:

$$\mathbf{y} = \mathcal{T}^{\alpha}\mathbf{x}, \quad \alpha \in \mathbb{R}^{+} \quad \Rightarrow \quad y(t) = x(\alpha t)$$

$\alpha > 1$ is a "compression", $0 < \alpha < 1$ is a "dilation":







Real-world frequency

frequency: number of repetitions per *second*

- f expressed in Hz (1/sec)
- alternatively, angular frequency in rad/s: $\Omega = 2\pi f$
- period for periodic signals is $T = \frac{1}{f} = \frac{2\pi}{\Omega}$ seconds

Fourier analysis: the CTFT

- in discrete time angular frequencies are limited to $[-\pi, \pi]$
- in continuous time the frequency range is infinite: $f \in \mathbb{R}$
- the continuous-time Fourier transform measures the similarity between a signal and *all* possible sinusoidal components:

$$\begin{aligned} X(f) &= \langle e^{j2\pi ft}, x(t) \rangle \\ &= \int_{-\infty}^{\infty} x(t) e^{-j2\pi ft} dt \qquad \leftarrow \textbf{not } \textit{a periodic function of } f \end{aligned}$$

- synthesis formula:

$$x(t) = \int_{-\infty}^{\infty} X(f) e^{j2\pi ft} df$$

Fourier analysis (in rad/s)

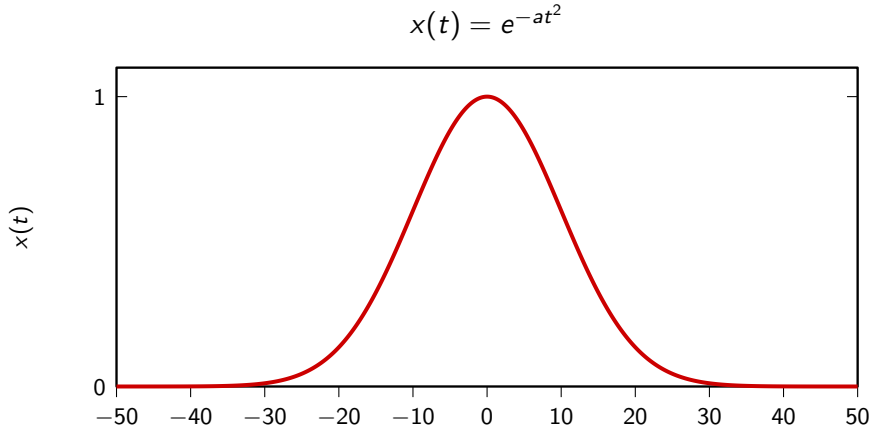
- alternate notation for analysis and synthesis formulas using rad/s

$$\begin{aligned} X(j\Omega) &= \langle e^{j\Omega t}, x(t) \rangle \\ &= \int_{-\infty}^{\infty} x(t) e^{-j\Omega t} dt \qquad \leftarrow \textit{not periodic} \end{aligned}$$

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} X(j\Omega) e^{j\Omega t} d\Omega$$

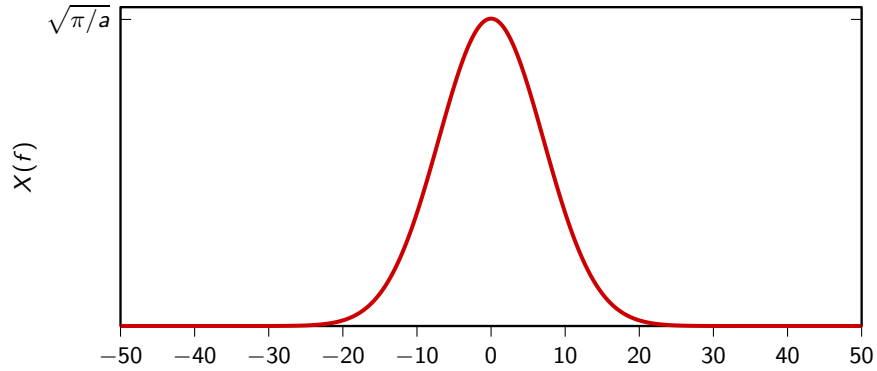
- if $X(s)$ is the Laplace transform of $x(t)$, the Fourier transform is $X(s)$ computed on the imaginary axis (just like the DTFT is the z-transform computed on the unit circle)

Example



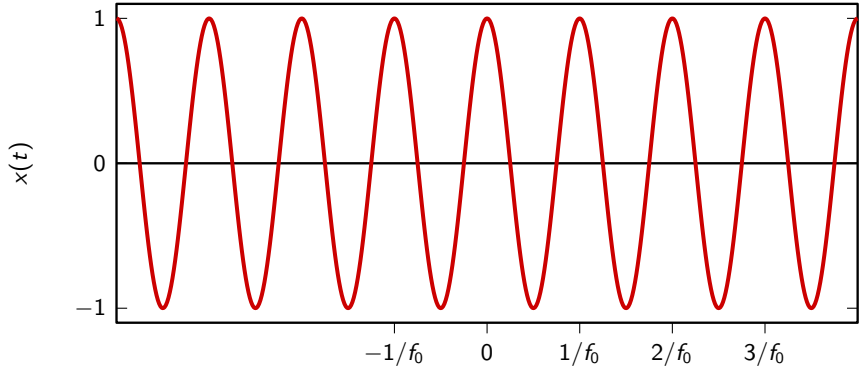
Example

$$X(f) = \sqrt{\pi/a}\, e^{-\frac{\pi^2}{a} f^2}$$



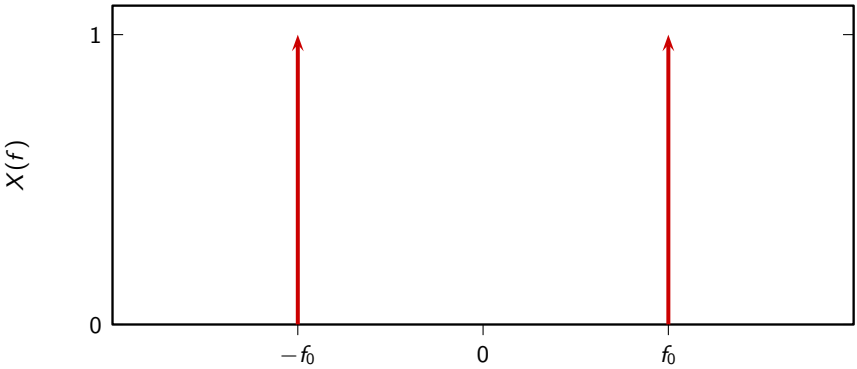
Example

$$x(t) = \cos(2\pi f_0 t)$$



Example

$X(f) = \delta(f \pm f_0)/2$ (note that this is not a periodized Dirac delta)



Important properties of the CTFT

- CTFT of time-shifted signals ($\tau \in \mathbb{R}$)

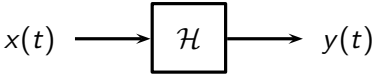
$$x(t-\tau) \overset{\text{CTFT}}{\longleftrightarrow} e^{-j2\pi\tau f} X(f)$$

- CTFT of time-scaled signals ($\alpha \in \mathbb{R}^+$):

$$x(\alpha t) \overset{\text{CTFT}}{\longleftrightarrow} \frac{1}{\alpha} X\left(\frac{f}{\alpha}\right)$$

- usual time-frequency duality:
  - $\alpha > 1$: signal is compressed, spectrum is stretched
  - $0 < \alpha < 1$: signal is stretched, spectrum is compressed

Continuous-time LTI systems



$$
\begin{aligned}
y(t) &= (\mathbf{h} * \mathbf{x})(t) \\
&= \int_{-\infty}^{\infty} x(\tau) h(t-\tau) d\tau
\end{aligned}
$$

Convolution theorem

$$Y(f) = X(f)\,H(f)$$

bandlimited signals

A new concept: bandlimited signals

a finite-energy, continuous-time signal $x(t)$ is called *bandlimited* if its spectrum has finite support:

$$X(f) = 0 \quad \text{for } f \notin [f_{\text{min}}, f_{\text{max}}]$$

The space of F_s-BL signals

a finite-energy, continuous-time signal $x(t)$ is called F_s-bandlimited if there exists a positive frequency value F_s such that

$$X(f) = 0 \quad \text{for } |f| > F_s/2$$

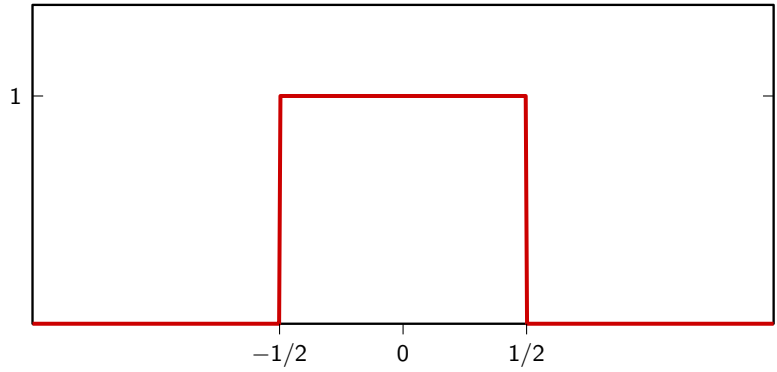
F_s is called the *total bandwidth* of the signal

Let's think of a prototypical bandlimited spectrum

- real and symmetric around $f = 0$ (so signal is real-valued)
- unit bandwidth (1 Hz)
- unit energy
- constant spectral amplitude over bandwidth

$$\Phi(f) = \text{rect}(f)$$

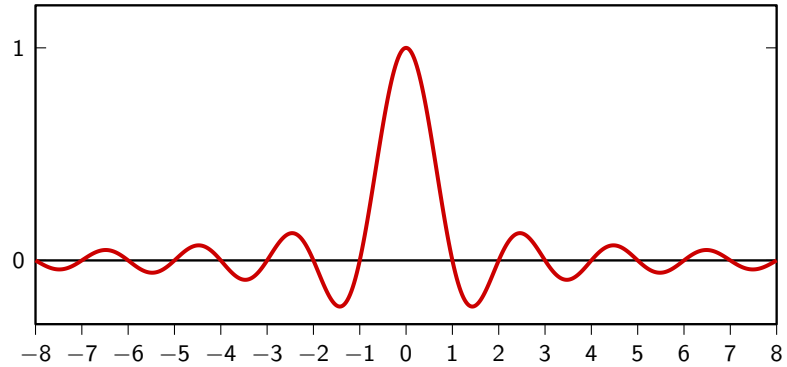
Prototypical bandlimited spectrum



The prototypical bandlimited signal

$$
\begin{aligned}
\varphi(t) &= \int_{-\infty}^{\infty} \Phi(f) e^{j2\pi ft} df \\
&= \int_{-1/2}^{1/2} e^{j2\pi ft} df \\
&= \frac{1}{j2\pi t}[e^{j\pi t} - e^{-j\pi t}] \\
&= \frac{\sin(\pi t)}{\pi t} \\
&= \mathrm{sinc}(t)
\end{aligned}
$$

The prototypical bandlimited signal



The prototypical bandlimited pair

$$\text{sinc}(t) \overset{\text{CTFT}}{\longleftrightarrow} \text{rect}(f)$$

Changing the bandwidth

if we time-stretch the sinc:

$$\text{sinc}(\alpha t) \xleftrightarrow{\text{CTFT}} \frac{1}{\alpha}\,\text{rect}\left(\frac{f}{\alpha}\right)$$

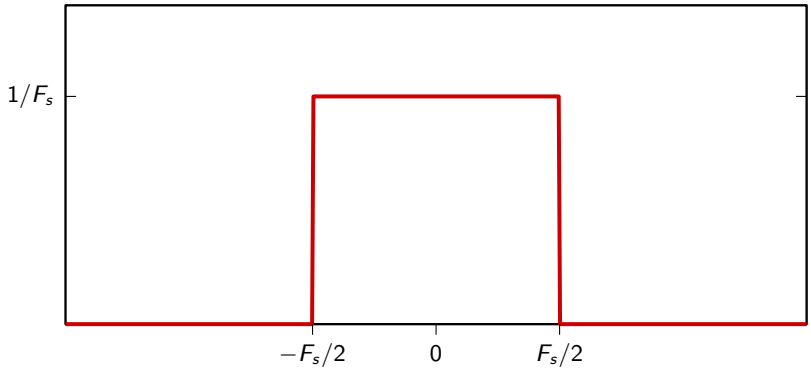
Changing the bandwidth

- pick a bandwith value F_s (Hz); let $T_s = 1/F_s$ seconds:

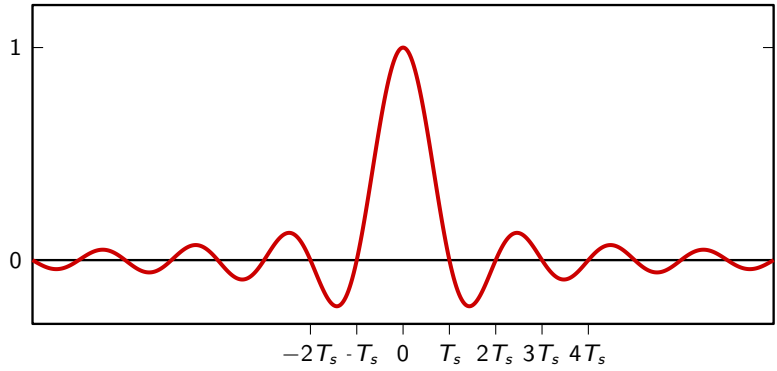
$$\text{sinc}\left(\frac{t}{T_s}\right) \overset{\text{CTFT}}{\longleftrightarrow} \frac{1}{F_s}\,\text{rect}\left(\frac{f}{F_s}\right)$$

- highest positive frequency is $F_s/2$, total bandwidth is F_s
- total energy is $1/F_s = T_s$

The prototypical F_s-BL spectrum



The prototypical F_s-BL signal



$(T_s = 1/F_s)$

interpolation

Overview:

- Polynomial interpolation
- Local interpolation
- Sinc interpolation

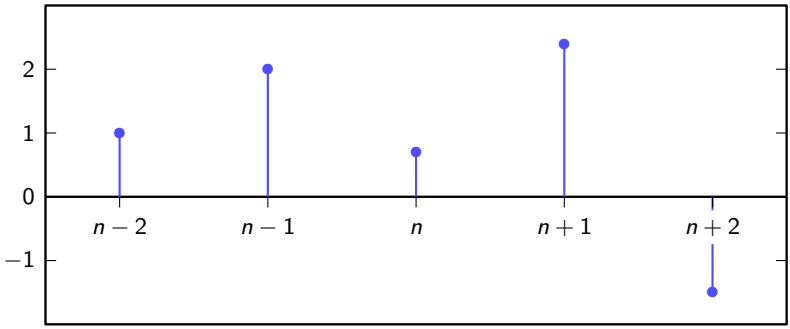
Interpolation

$$x[n] \longrightarrow x(t)$$

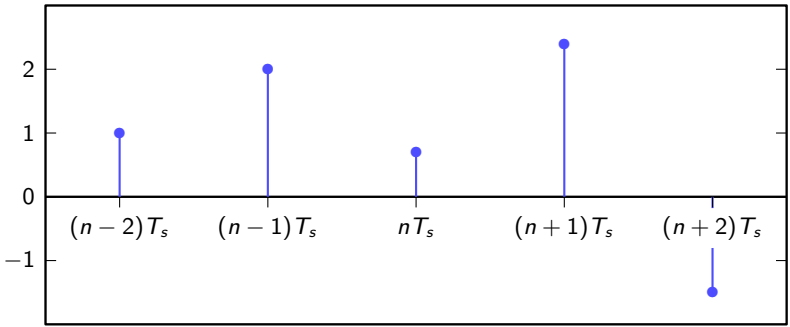
use a new sample every T_s seconds

"fill the gap" between successive samples

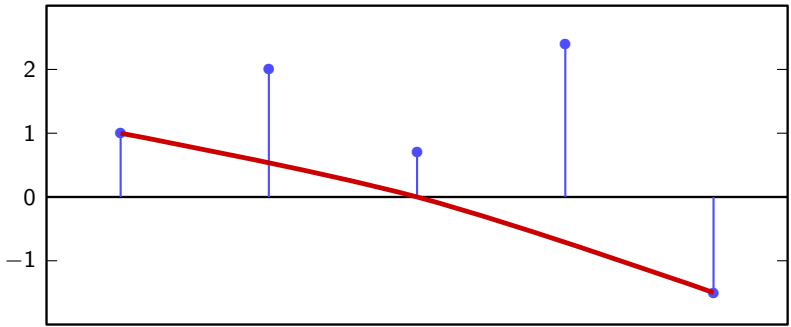
Example



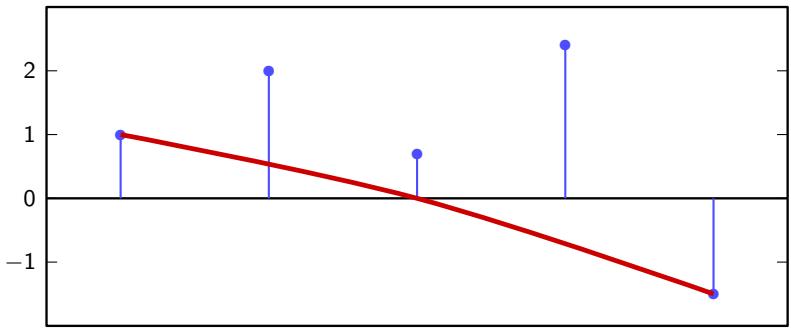
Example



Is this a good interpolation?

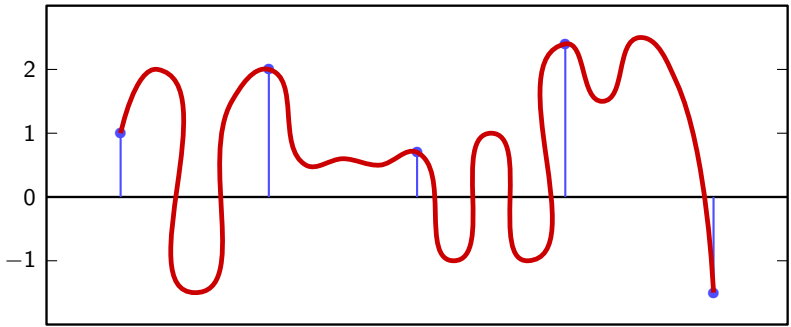


Is this a good interpolation?

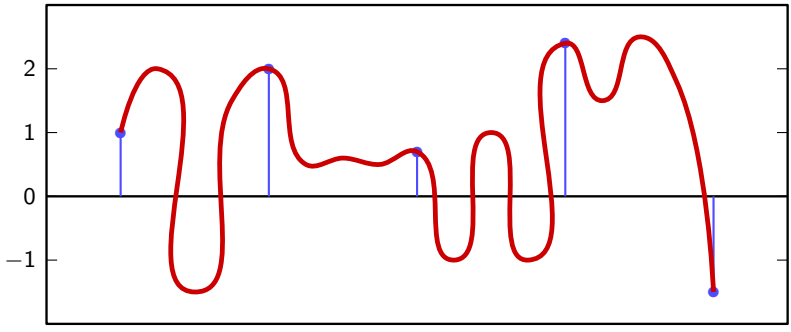


not really, the interpolation doesn't go through all data points...

Is this a good interpolation?

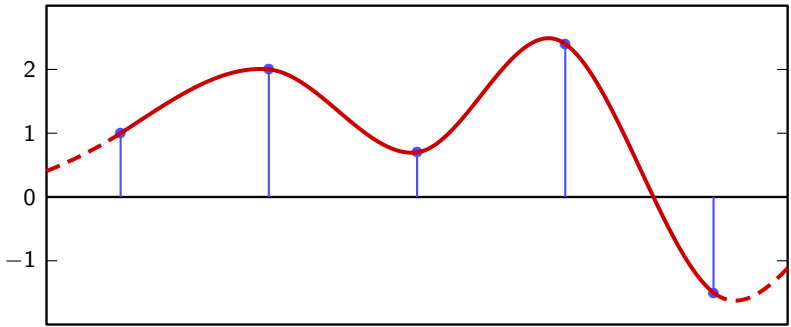


Is this a good interpolation?

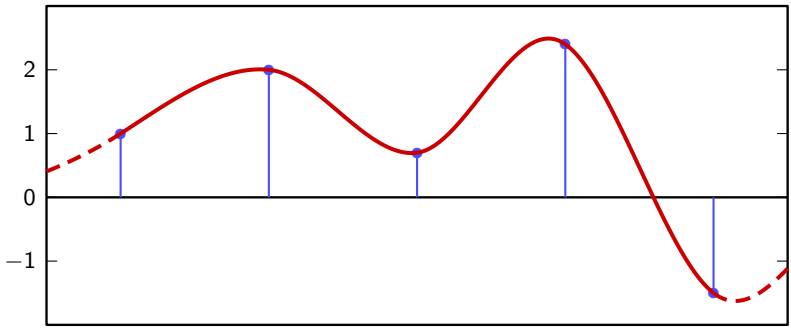


not really, the interpolation seems to “move around” too much...

What about this?



What about this?



this looks pretty convincing; why?

Interpolation requirements

- decide on the timebase T_s
- make sure $x(nT_s) = x[n]$
- make sure $x(t)$ is *smooth*

Choosing the timebase

- T_s is the "spacing" between interpolation points in seconds
- without loss of generality we can choose $T_s = 1$
- if $x_c(t)$ solves the interpolation problem for $T_s = 1$, then $x(t) = x_c(t/T)$ solves it for $T_s = T$:
  - $x(nT) = x_c(nT/T) = x_c(n) = x[n]$
  - if $x_c(t)$ is smooth, so is $x_c(t/T)$

Why smoothness?

- jumps (1st order discontinuities) would require the signal to move "faster than light"...
- 2nd order discontinuities would require infinite acceleration
- ...
- the interpolation should be infinitely differentiable
- "natural" solution: polynomial interpolation

Polynomial interpolation

Polynomial interpolation

- $N+1$ points $\rightarrow$ polynomial of degree N
- $x_c(t) = a_0 + a_1 t + a_2 t^2 + \ldots + a_N t^N$
- straightforward approach ($T_s = 1$):

$$\begin{cases} x_c(0) = x[0] \\ x_c(1) = x[1] \\ x_c(2) = x[2] \\ \qquad \ldots \\ x_c(N) = x[N] \end{cases}$$

Polynomial interpolation

Without loss of generality:

- even polynomial degree $2N$
- fit polynomial over $2N+1$ data points symmetrically distributed around zero

$$\begin{cases} x_c(-N) = x[-N] \\ x_c(-N+1) = x[-N+1] \\ \dots \\ x_c(0) = x[0] \\ \dots \\ x_c(N-1) = x[N-1] \\ x_c(N) = x[N] \end{cases}$$

Polynomial interpolation

- find the $2N+1$ coefficients $a_0, \ldots, a_{2N}$ by solving:

$$\sum_{k=0}^{2N} a_k n^k = x[n] \qquad n = -N, -N+1, \ldots, 0, \ldots, N-1, N$$

Lagrange interpolation

Let's use the power of vector spaces:

- P_N: space of degree-$2N$ polynomials over $I_N = [-N, N]$
- interpolation will be a linear combination of basis vectors for P_N
- what is a good basis for *interpolation?*

Lagrange interpolation

- P_N: space of degree-$2N$ polynomials over $I_N = [-N, N]$
- a basis for P_N is the family of $2N + 1$ Lagrange polynomials

$$L_n^{(N)}(t) = \prod_{\substack{k=-N \\ k \neq n}}^{N} \frac{t-k}{n-k} \qquad n = -N, \ldots, N$$

Lagrange polynomials for l_2

$$L_{-2}^{(2)}(t) = \left(\frac{t+1}{-2+1}\right)\left(\frac{t}{-2}\right)\left(\frac{t-1}{-2-1}\right)\left(\frac{t-2}{-2-2}\right) = \frac{(t+1)t(t-1)(t-2)}{24}$$

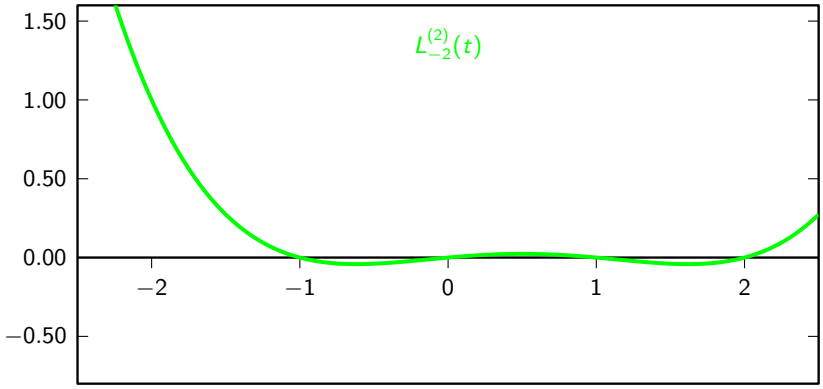
$$L_{-1}^{(2)}(t) = \left(\frac{t+2}{-1+2}\right)\left(\frac{t}{-1}\right)\left(\frac{t-1}{-1-1}\right)\left(\frac{t-2}{-1-2}\right) = \frac{(t+2)t(t-1)(t-2)}{-6}$$

$$L_{0}^{(2)}(t) = \left(\frac{t+2}{2}\right)\left(\frac{t+1}{1}\right)\left(\frac{t-1}{-1}\right)\left(\frac{t-2}{-2}\right) = \frac{(t+2)(t+1)(t-1)(t-2)}{-4}$$

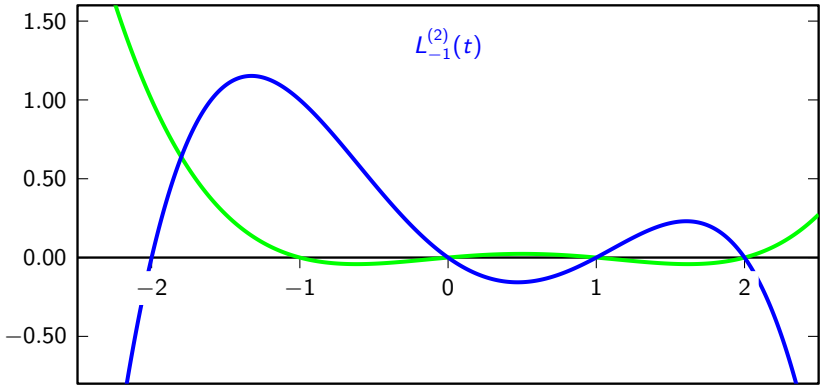
$$L_{1}^{(2)}(t) = L_{-1}^{(2)}(-t)$$

$$L_{2}^{(2)}(t) = L_{-2}^{(2)}(-t)$$

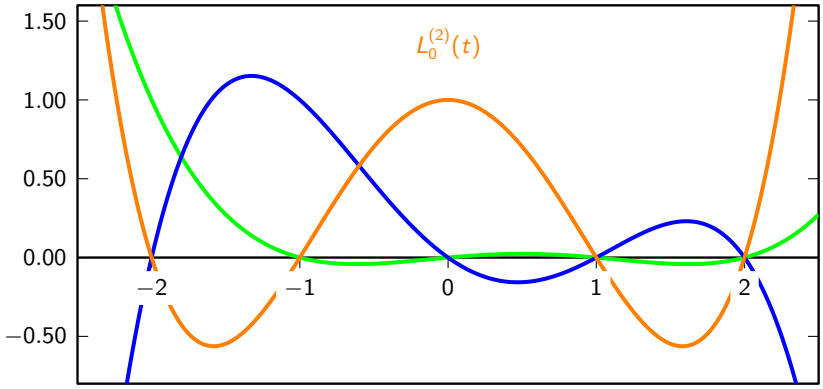
Lagrange interpolation polynomials



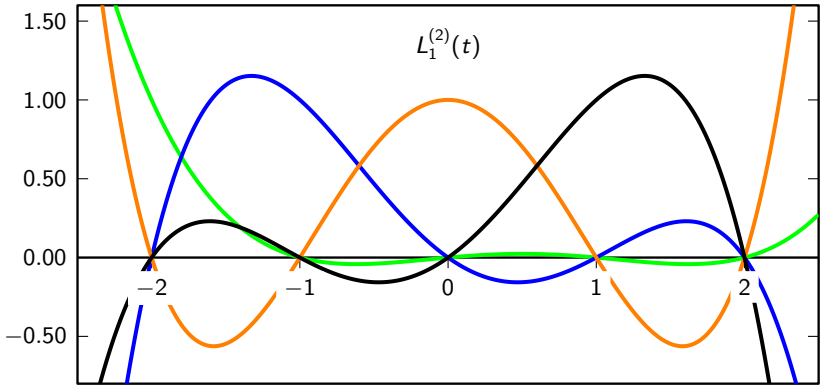
Lagrange interpolation polynomials



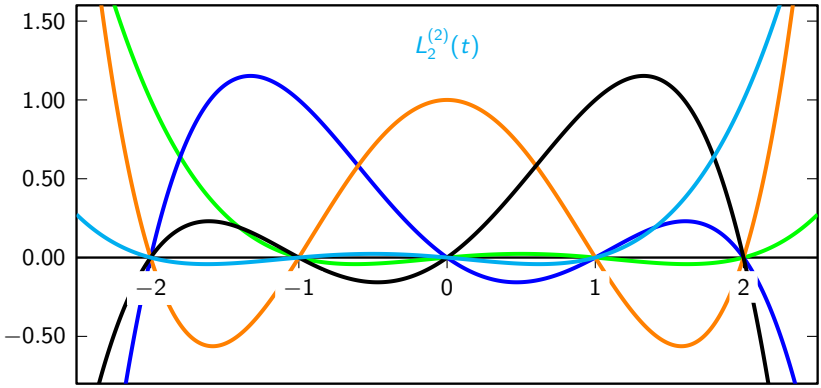
Lagrange interpolation polynomials



Lagrange interpolation polynomials



Lagrange interpolation polynomials



Interpolation property of Lagrange polynomials

$$m \in \mathbb{N} \quad \Rightarrow \quad L_n^{(N)}(m) = \begin{cases} 1 & \text{if } n = m \\ 0 & \text{if } n \neq m \end{cases} \qquad -N \leq n, m \leq N$$

Lagrange interpolator for the data set

$$x_c(t) = \sum_{n=-N}^{N} x[n] L_n^{(N)}(t)$$

Lagrange interpolation

The Lagrange interpolator *is* the unique polynomial interpolation for the data set:

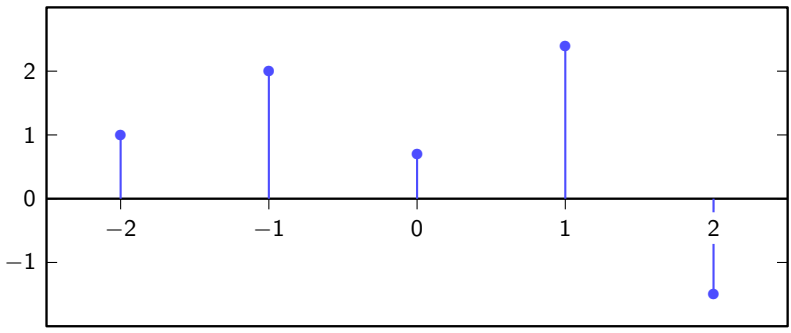
- a polynomial of degree $2N$ through $2N+1$ points is uniquely defined
- the Lagrangian interpolator satisfies

$$x_c(n) = x[n] \qquad \text{for } -N \leq n \leq N$$

since

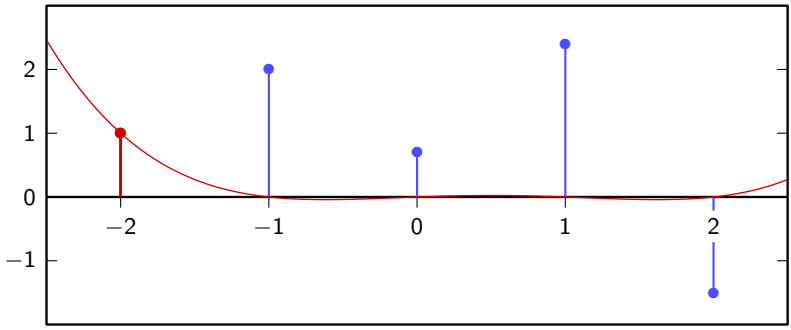
$$L_n^{(N)}(m) = \begin{cases} 1 & \text{if } n = m \\ 0 & \text{if } n \neq m \end{cases} \qquad -N \leq n, m \leq N$$

Lagrange interpolation

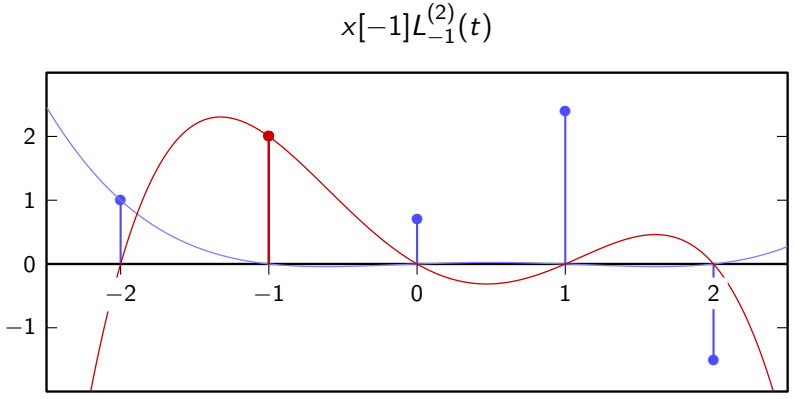


Lagrange interpolation

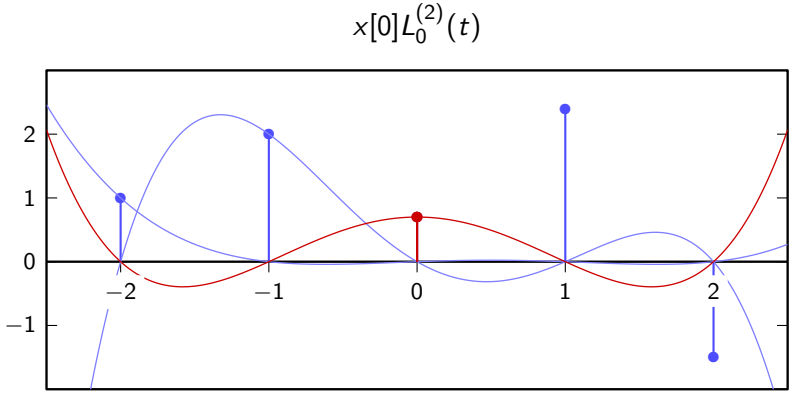
$x[-2]L_{-2}^{(2)}(t)$



Lagrange interpolation

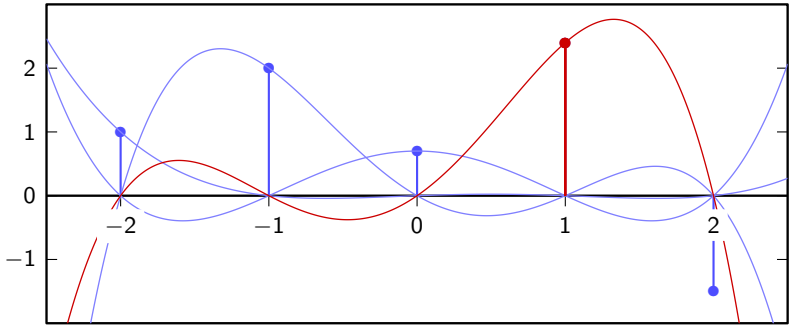


Lagrange interpolation

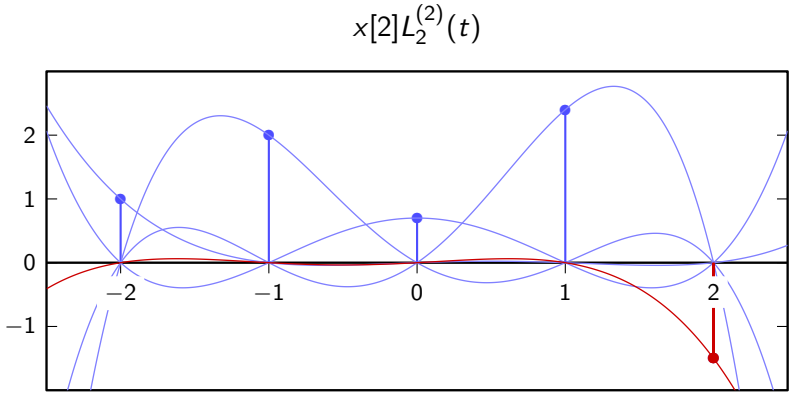


Lagrange interpolation

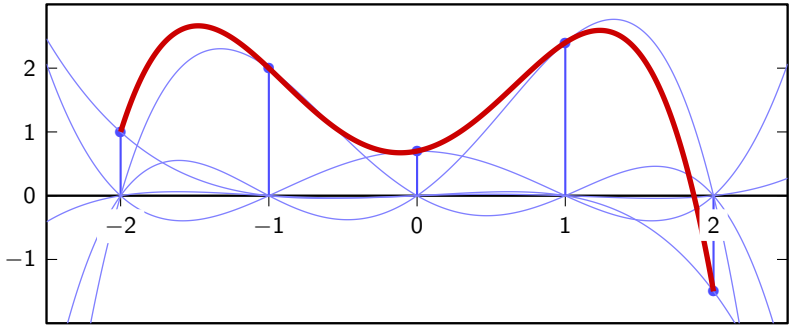
$x[1]L_1^{(2)}(t)$



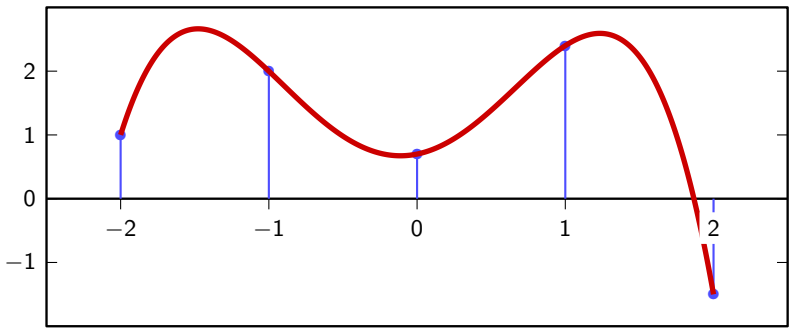
Lagrange interpolation



Lagrange interpolation



Lagrange interpolation



Polynomial interpolation

key property:

- maximally smooth (infinitely many continuous derivatives)

drawback:

- interpolation "machine" depend on N: we need to use a different set of polynomials if the length of the dataset changes

can we find a "universal" interpolation machine?

Relaxing the interpolation requirements

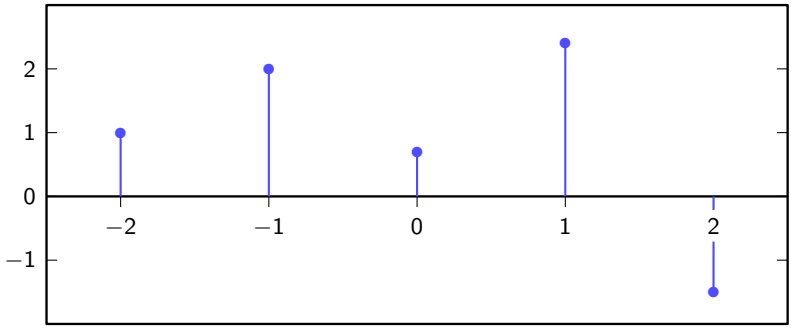
- make sure $x_c(n) = x[n]$
- make sure $x_c(t)$ is *smooth*

Relaxing the interpolation requirements

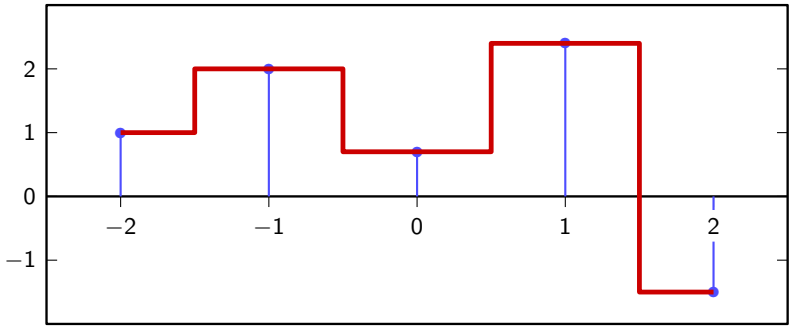
- make sure $x_c(n) = x[n]$
- make sure $x_c(t)$ is *smooth*

Local interpolation

Zero-order interpolation



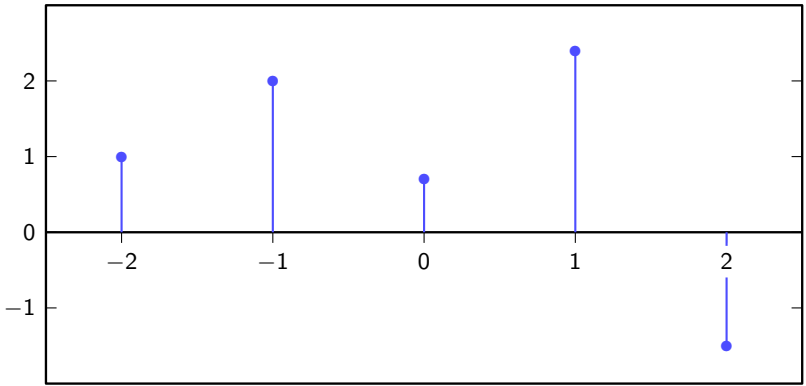
Zero-order interpolation



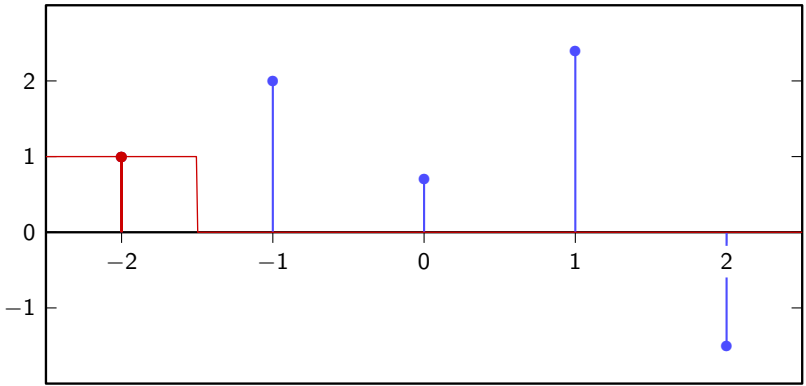
Zero-order interpolation

- $x_0(t) = x[\,\lfloor t + 0.5 \rfloor\,], \qquad -N \le t \le N$
- $x_0(t) = \sum_{n=-N}^{N} x[n]\,\mathrm{rect}(t - n)$
- interpolation kernel: $i_0(t) = \mathrm{rect}(t)$
- $i_0(t)$: "zero-order hold"
- interpolator's support is 1
- interpolation is not even continuous

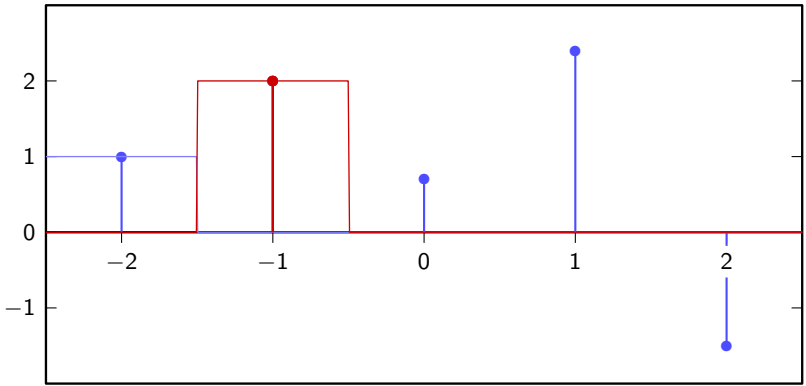
Zero-order interpolation



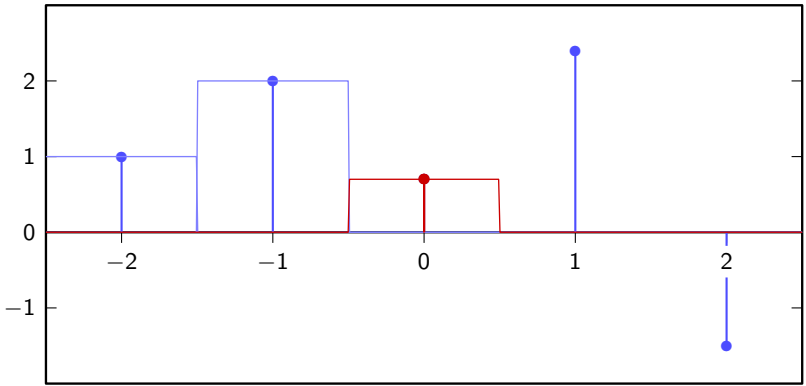
Zero-order interpolation



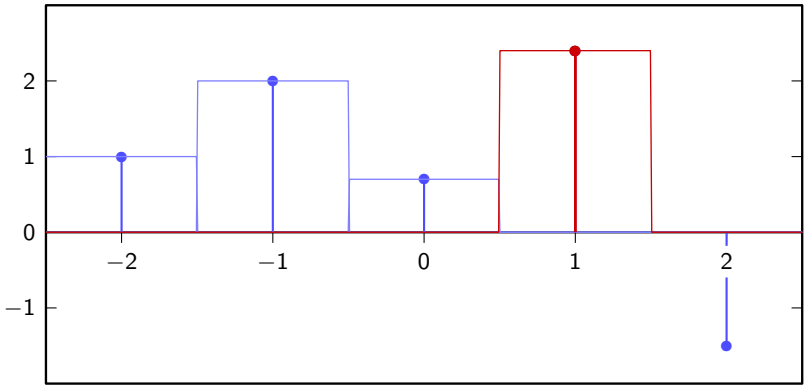
Zero-order interpolation



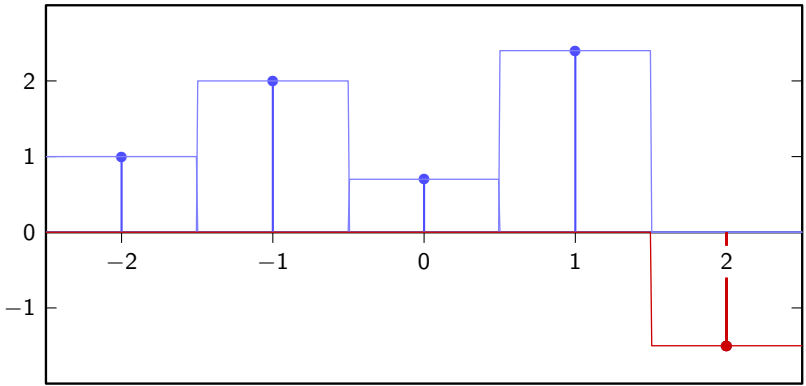
Zero-order interpolation



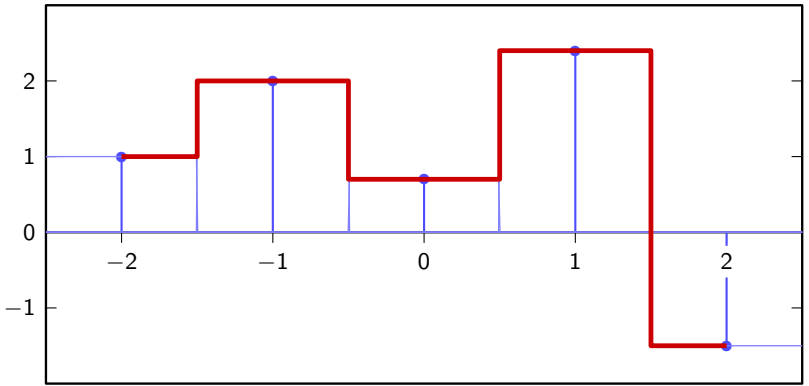
Zero-order interpolation



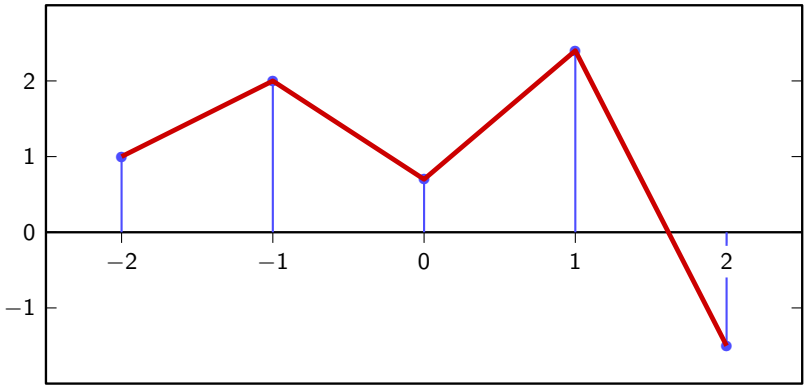
Zero-order interpolation



Zero-order interpolation



First-order interpolation



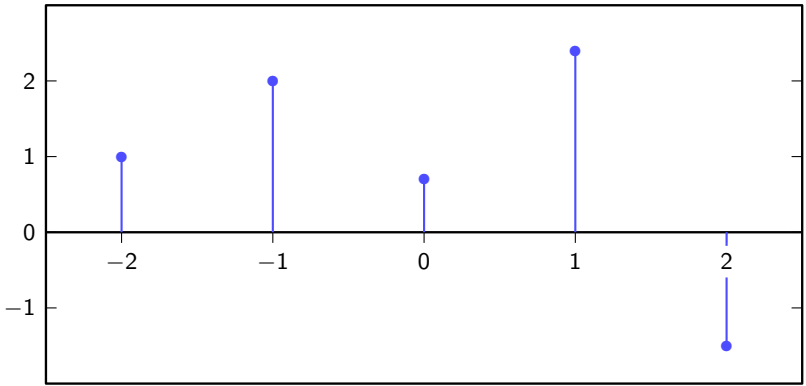
First-order interpolation

- "connect the dots" strategy
- $x_1(t) = \sum_{n=-N}^{N} x[n]\, i_1(t-n)$
- interpolation kernel:

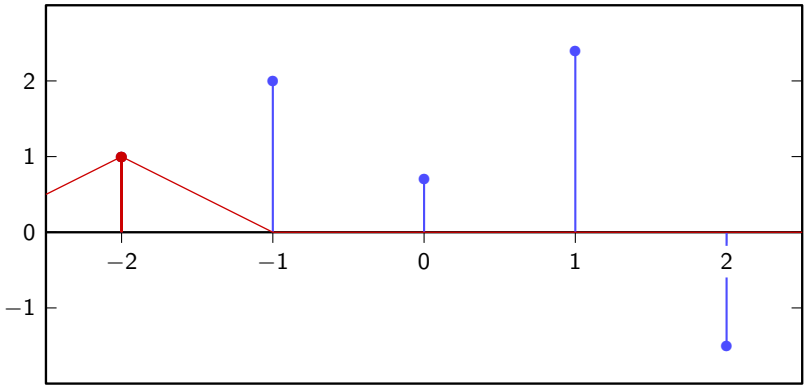
$$i_1(t) = \begin{cases} 1-|t| & |t| \le 1 \\ 0 & \text{otherwise} \end{cases}$$

- interpolator's support is 2
- interpolation is continuous but derivative is not

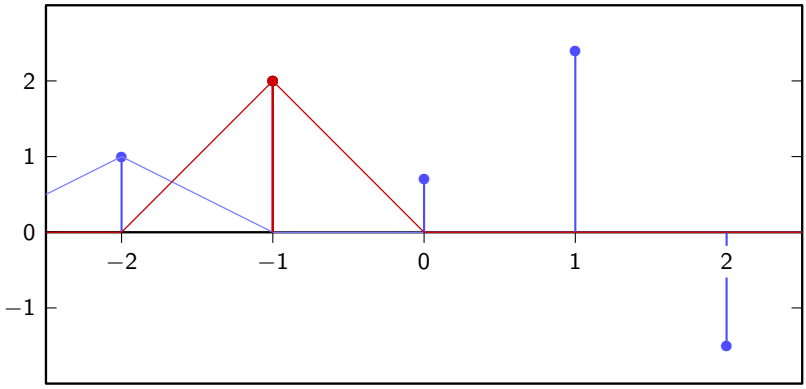
First-order interpolation



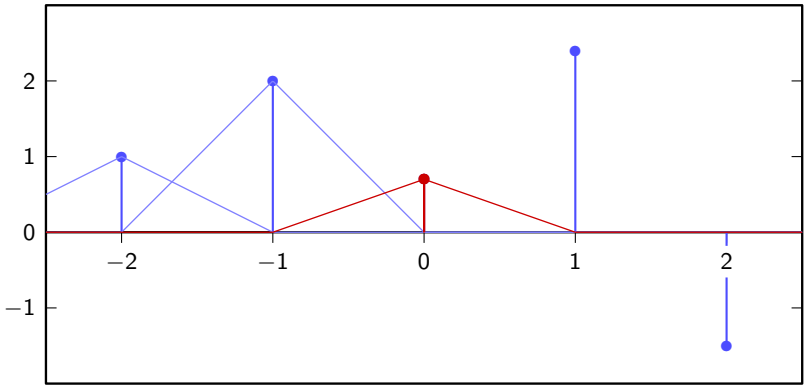
First-order interpolation



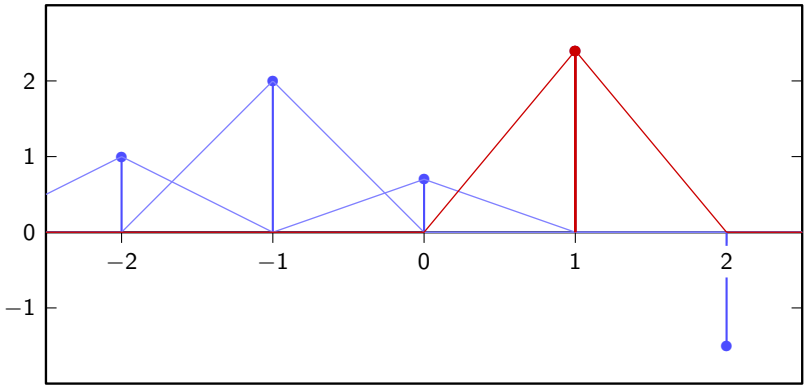
First-order interpolation



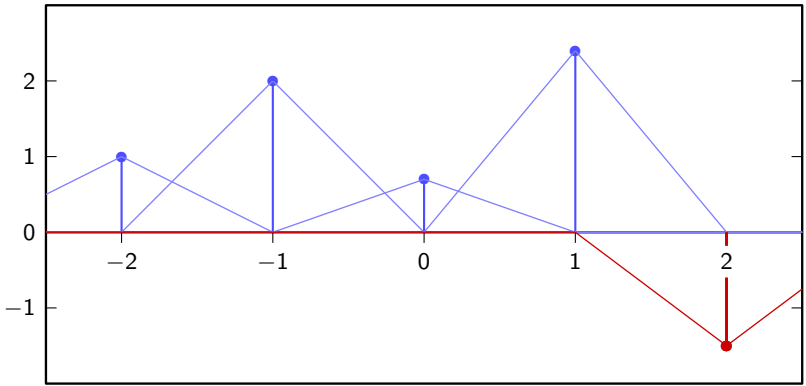
First-order interpolation



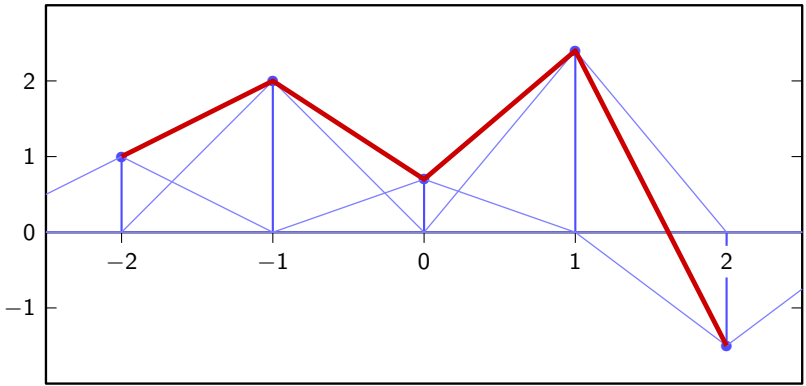
First-order interpolation



First-order interpolation



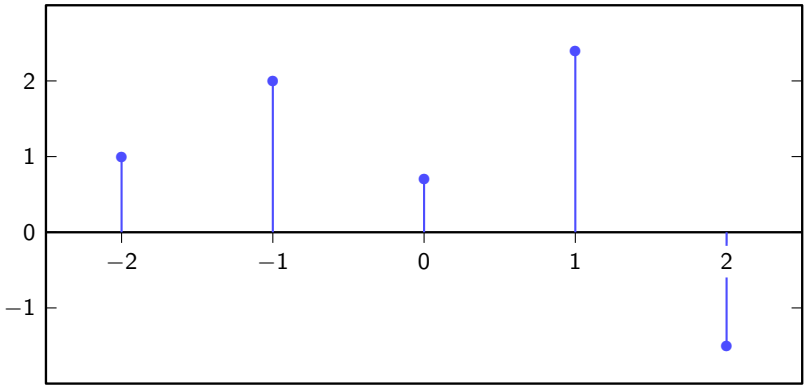
First-order interpolation



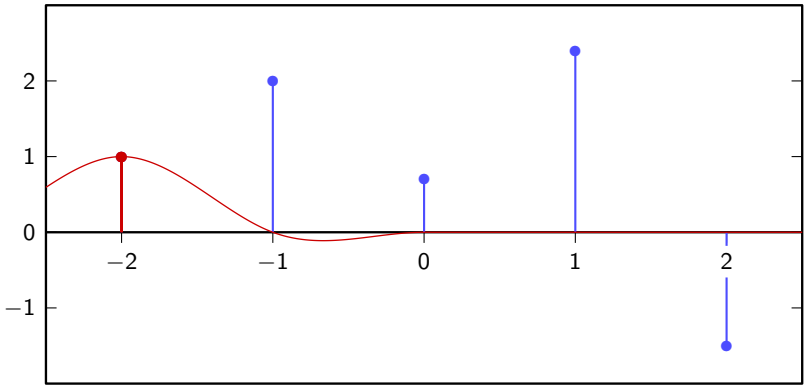
Third-order interpolation

- $x_3(t) = \sum_{n=-N}^{N} x[n]\, i_3(t-n)$
- interpolation kernel obtained by splicing two cubic polynomials
- interpolator's support is 4
- interpolation is continuous up to second derivative

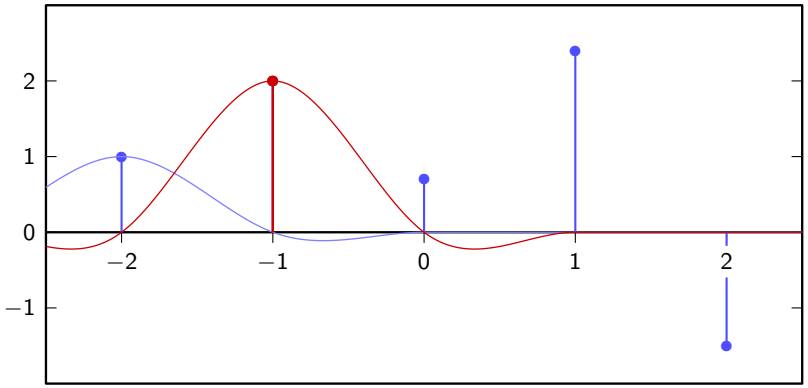
Third-order interpolation



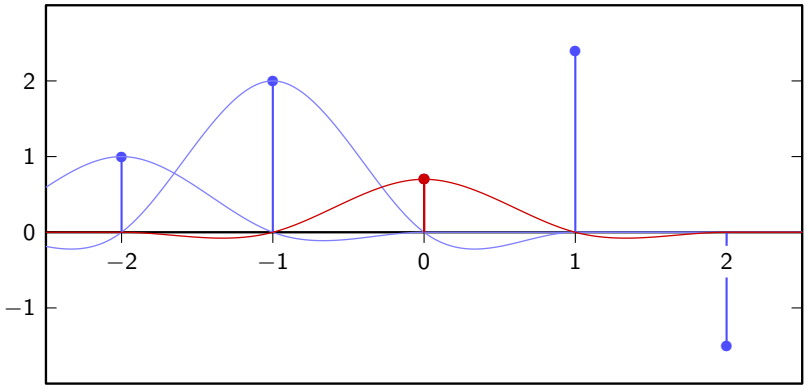
Third-order interpolation



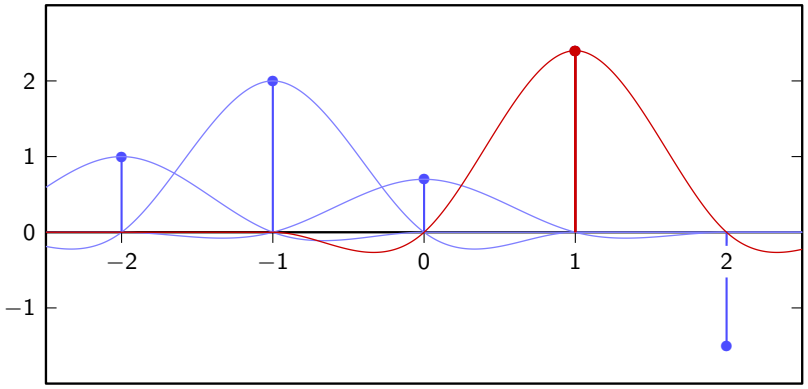
Third-order interpolation



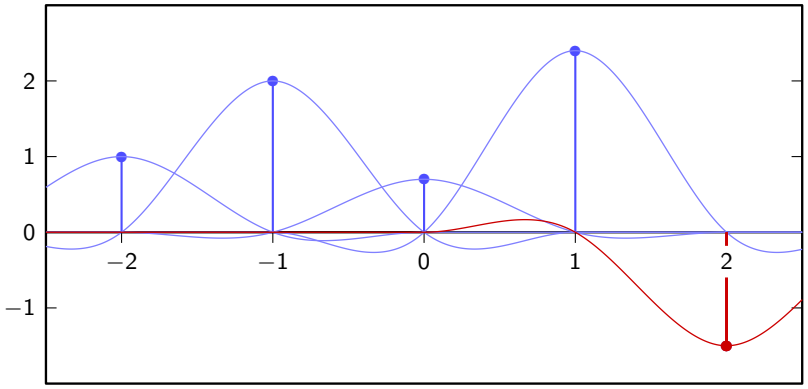
Third-order interpolation



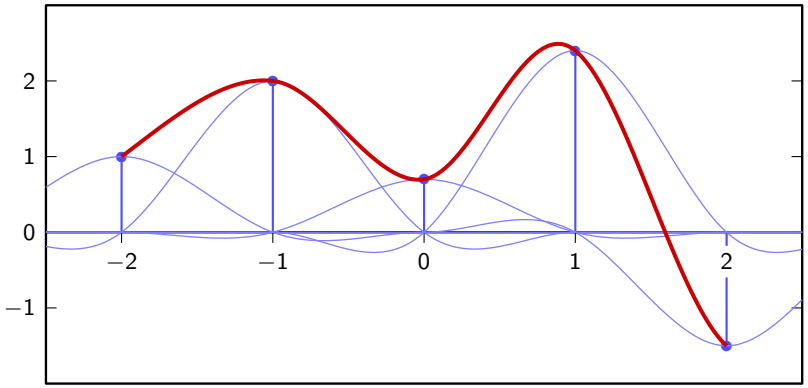
Third-order interpolation



Third-order interpolation



Third-order interpolation



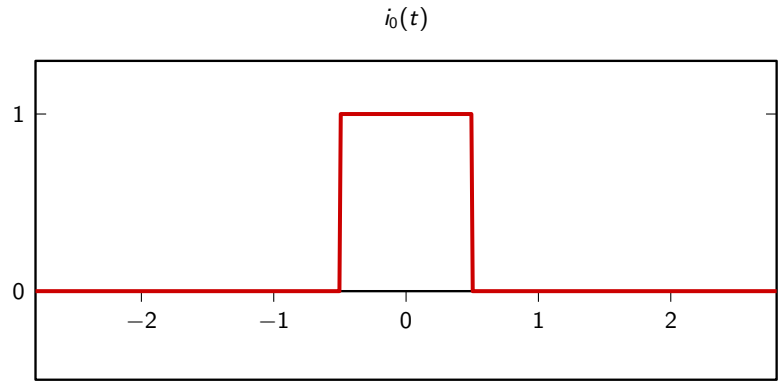
Local interpolation schemes

$$x_c(t) = \sum_{n=-N}^{N} x[n]\, i_c(t-n)$$

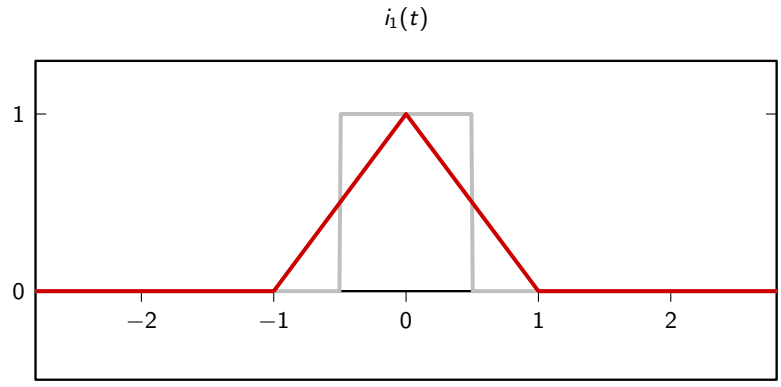
Kernel must satisfy the interpolation properties:

- $i_c(0) = 1$
- $i_c(m) = 0$ for m a nonzero integer.

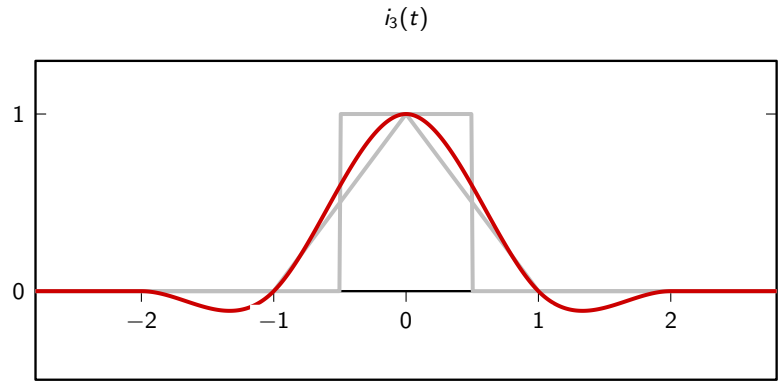
Local interpolators



Local interpolators



Local interpolators



Local interpolation

key property:

- same interpolating function independently of N

drawback:

- lack of smoothness

Polynomial interpolation

key property:

- maximally smooth (infinitely many continuous derivatives)

drawback:

- interpolation kernels depend on N

Sync interpolation

A remarkable result:

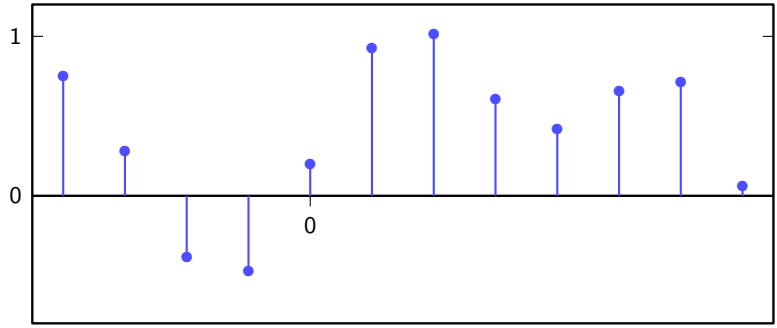
$$\lim_{N\to\infty} L_n^{(N)}(t) = \operatorname{sinc}(t-n)$$

- in other words: $\lim_{N\to\infty} L_n^{(N)}(t) = L_0^{(\infty)}(t-n)$
- in the limit, local and global interpolation are the same!

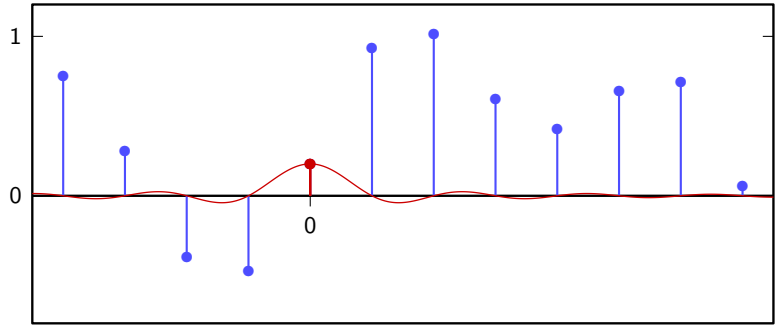
Sinc interpolation formula

$$x_c(t) = \sum_{n=-\infty}^{\infty} x[n] \, \mathrm{sinc}\,(t - n)$$

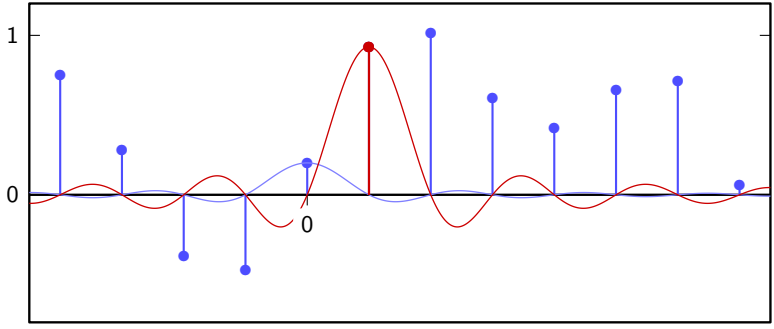
Sinc interpolation



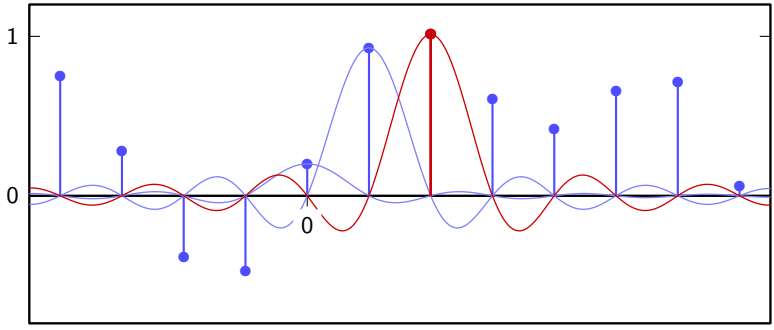
Sinc interpolation



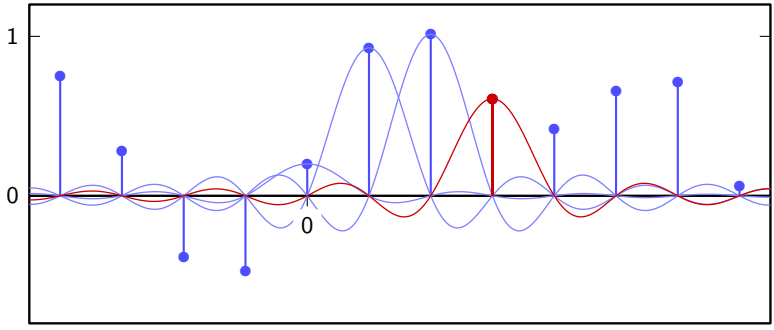
Sinc interpolation



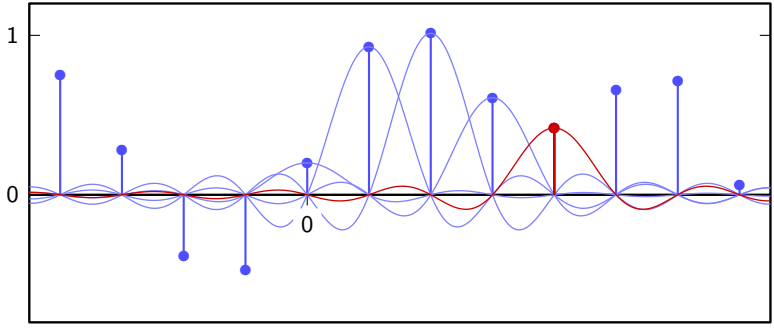
Sinc interpolation



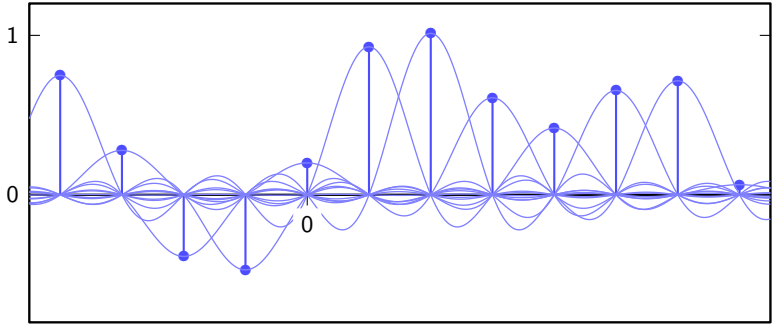
Sinc interpolation



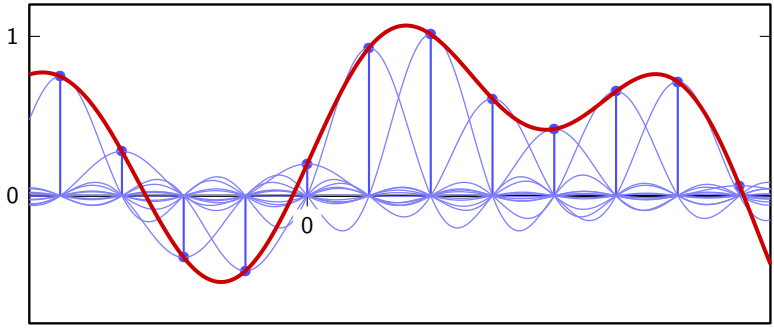
Sinc interpolation



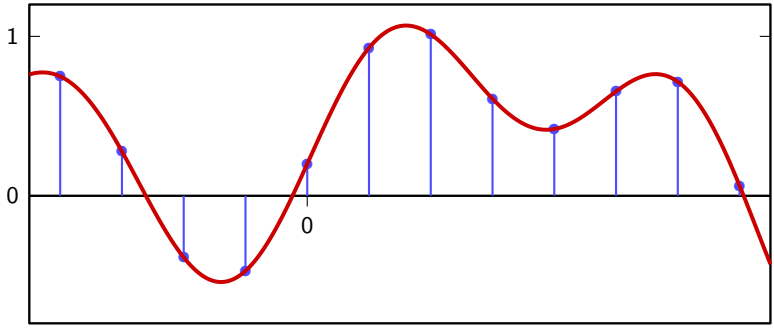
Sinc interpolation



Sinc interpolation



Sinc interpolation



Convergence: graphical "proof"

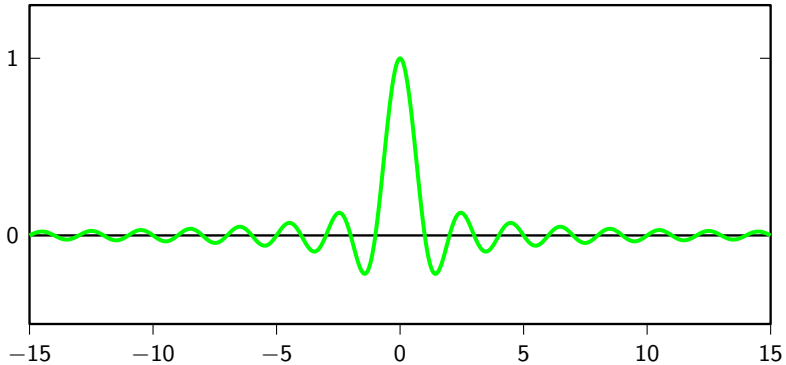
$$L_n^{(N)}(t) = \prod_{\substack{k=-N \\ k\neq n}}^{N} \frac{t-k}{n-k}$$

$$L_0^N(t) = \prod_{\substack{k=-N \\ k\neq 0}}^{N} \frac{t-k}{-k} = \prod_{k=-N}^{-1} \frac{t-k}{-k} \prod_{k=1}^{N} \frac{t-k}{-k}$$

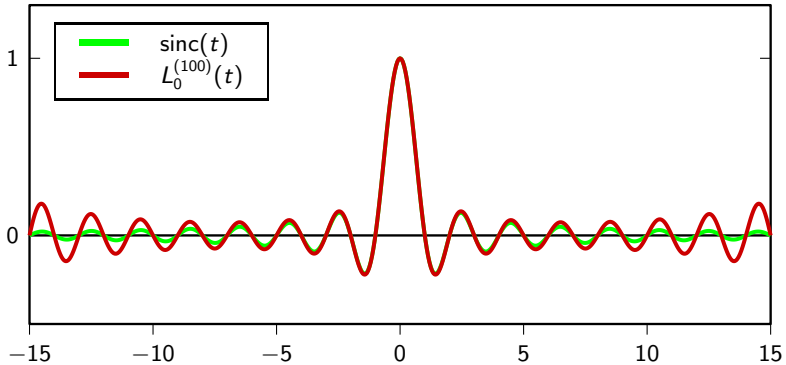
$$= \prod_{k=1}^{N} \frac{t+k}{k} \prod_{k=1}^{N} \frac{t-k}{-k}$$

$$= \prod_{k=1}^{N} \left(1 - \frac{t^2}{k^2}\right)$$

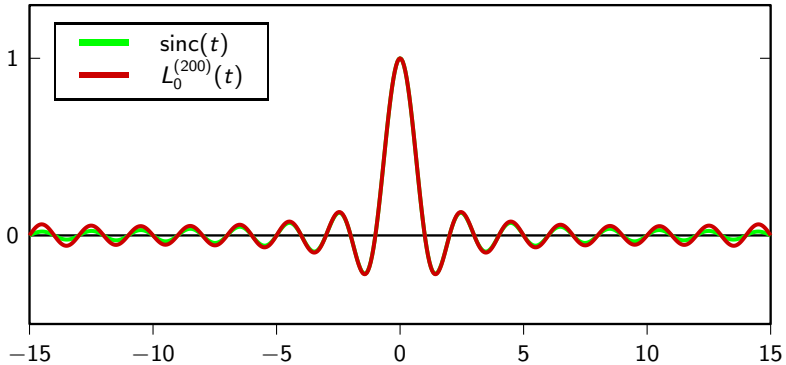
Convergence: graphical “proof”



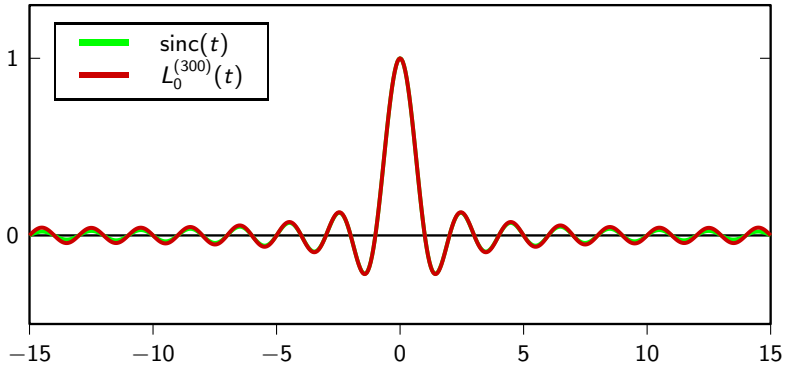
Convergence: graphical "proof"



Convergence: graphical "proof"



Convergence: graphical "proof"



Convergence: mathematical intuition

- $\mathrm{sinc}(t-n)$ and $L_n^{(\infty)}(t)$ share an infinite number of zeros:

$$\mathrm{sinc}(m-n) = \delta[m-n] \qquad m, n \in \mathbb{Z}$$

$$L_n^{(N)}(m) = \delta[m-n] \qquad m, n \in \mathbb{Z}, \quad -N \leq n, m \leq N$$

Convergence: Euler's "proof" (1748)

very cute (if non-rigorous) proof – see handout or book for details

Convergence: rigorous proof

uses the properties of Fourier series expansions – see handout or book for details

Sinc interpolation formula for any T_s

$$x_{T_s}(t) = x_c(t/T_s) = \sum_{n=-\infty}^{\infty} x[n] \operatorname{sinc}\left(\frac{t - nT_s}{T_s}\right)$$

the spectrum of sinc-interpolated signals

Sinc interpolation

- discrete-time signal $x[n]$, with DTFT $X(\omega)$
- sinc interpolation $x_c(t) = \sum_{n=-\infty}^{\infty} x[n]\,\mathrm{sinc}(t-n)$
- call $X_c(f)$ the continuous-time Fourier transform of the interpolation

what is the relationship between $X(\omega)$ and $X_c(f)$?

Spectral representation

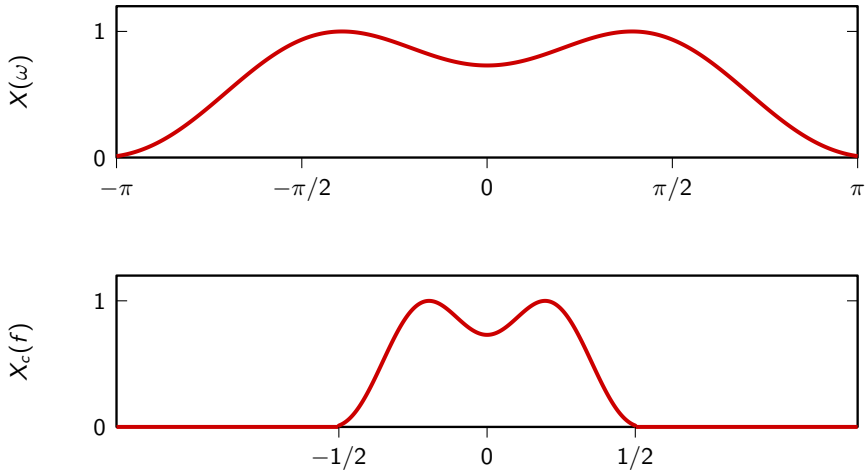
$$
\begin{aligned}
X_c(f) &= \int_{-\infty}^{\infty} x_c(t)\, e^{-j2\pi ft}\, dt \\
&= \int_{-\infty}^{\infty} \sum_{n=-\infty}^{\infty} x[n]\, \mathrm{sinc}(t-n) e^{-j2\pi ft}\, dt \\
&= \sum_{n=-\infty}^{\infty} x[n] \int_{-\infty}^{\infty} \mathrm{sinc}(t-n) e^{-j2\pi ft}\, dt \\
&= \sum_{n=-\infty}^{\infty} x[n]\, e^{-j(2\pi f)n}\ \mathrm{rect}(f) \\
&= \begin{cases} X(2\pi f) & |f| \le 1/2 \\ 0 & \text{otherwise} \end{cases}
\end{aligned}
$$

Spectral representation

$$X_c(f) = \begin{cases} X(2\pi f) & |f| \leq 1/2 \\ 0 & \text{otherwise} \end{cases}$$

- $X(2\pi f)$ is the DTFT scaled so that $\omega = \pi \rightarrow f = 1/2$ Hz
- $X(2\pi f)$ is periodic with period 1
- the rect keeps only the zero-centered period

Spectrum of interpolated signal



Changing the timebase

the signal $x_c(t)$ is obtained using an interpolation interval of one second ($T_s = 1$)
an interval of T_s seconds (i.e. an interpolation rate $F_s = 1/T_s$) yields the interpolation

$$x_{T_s}(t) = x_c(t/T_s) = \sum_{n=-\infty}^{\infty} x[n] \operatorname{sinc}\left(\frac{t - nT_s}{T_s}\right)$$

whose Fourier transform is

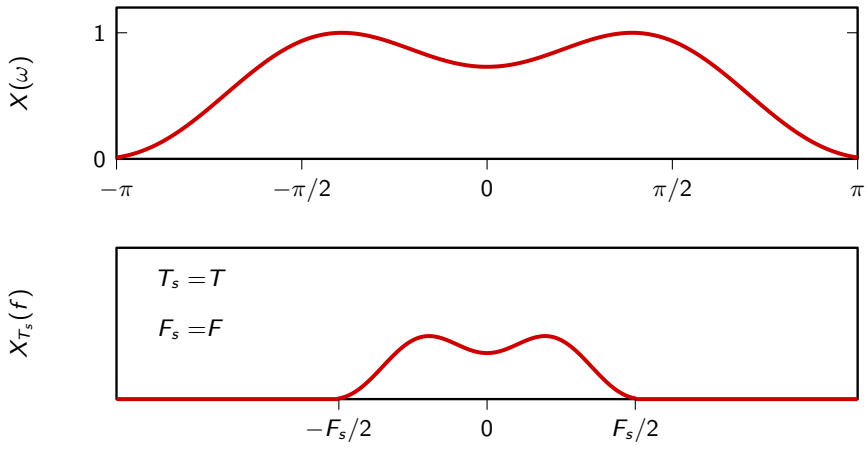
$$X_{T_s}(f) = T_s X_c(T_s f) = \frac{1}{F_s} X_c\left(\frac{f}{F_s}\right)$$

Spectrum of interpolated signals

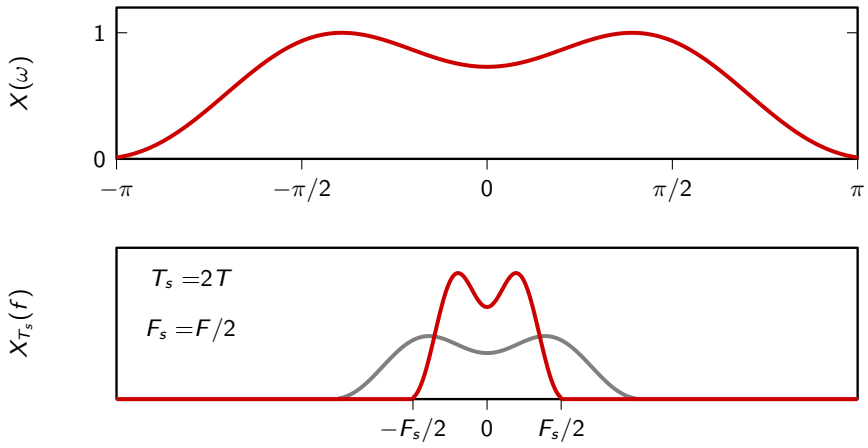
$$X_{T_s}(f) = \begin{cases} \frac{1}{F_s} X\left(2\pi \frac{f}{F_s}\right) & |f| \leq F_s/2 \\ 0 & \text{otherwise} \end{cases}$$

- $X(2\pi f/F_s)$ is the DTFT scaled so that $\omega = \pi \rightarrow f = F_s/2$ Hz
- $X(2\pi f/F_s)$ is periodic with period F_s
- the rect keeps only the zero-centered period
- the spectrum is scaled in amplitude by $1/F_s$

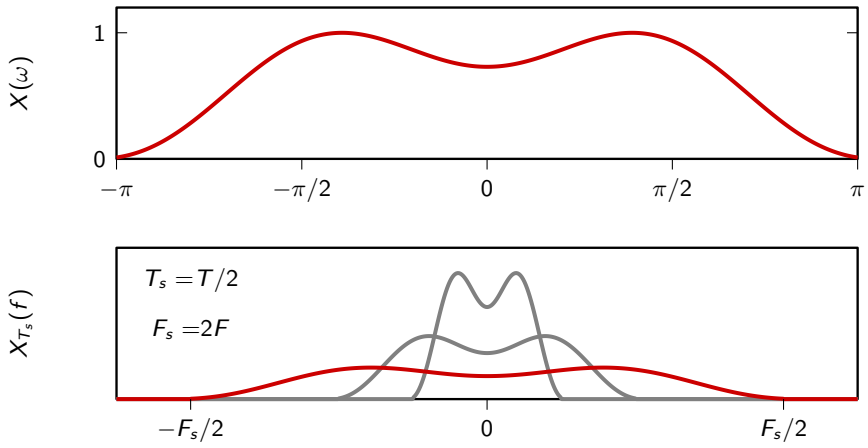
Spectrum of interpolated signals



Spectrum of interpolated signals



Spectrum of interpolated signals



Spectrum of interpolated signals

for an interpolation rate $F_s = 1/T_s$:

- $X(f)$ will be F_s-bandlimited
- fast interpolation (T_s small) $\rightarrow$ wide spectrum
- slow interpolation (T_s large) $\rightarrow$ narrow spectrum
- (for those who remember...) it's like spinning a vynil record faster or slower, changing the sound