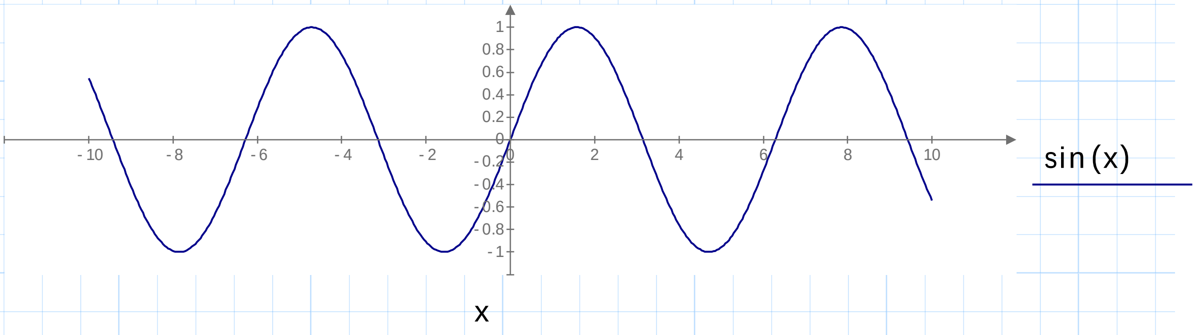
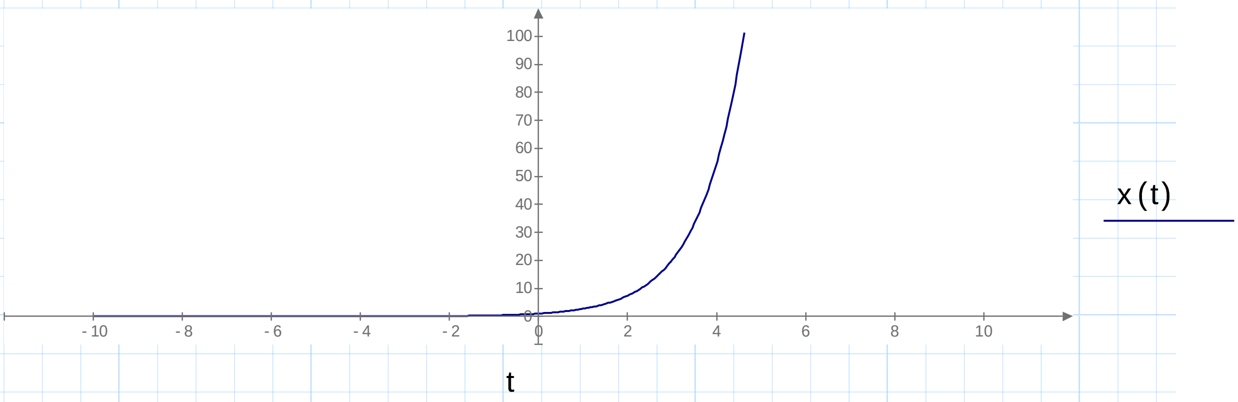
Signals and Systems Using Mathcad (Tutorial) by Derose and Veronis.
Chapter 2 Time Domain Analysis
Entered by: Karl S Bogha Dhaliwal
Part 1 of 2

Example 1 continous signal



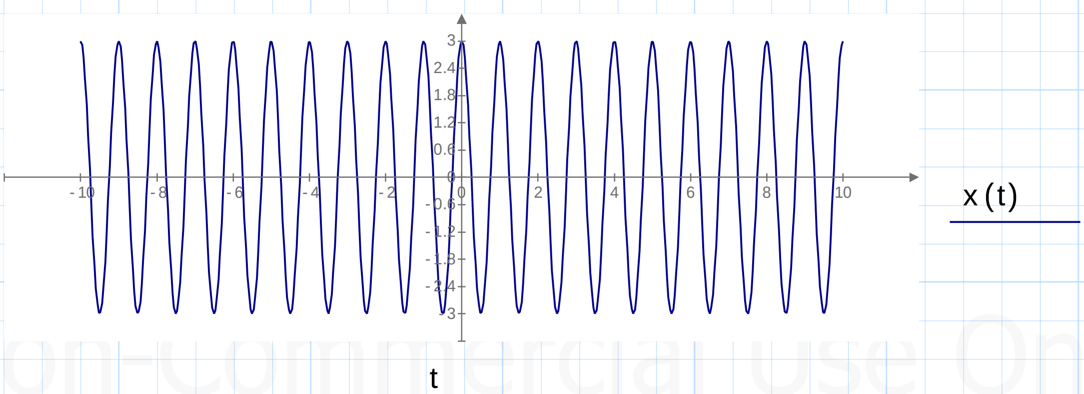
Example 2 continous signal

$x(t) := \exp(t)$



Example 3 continous signal

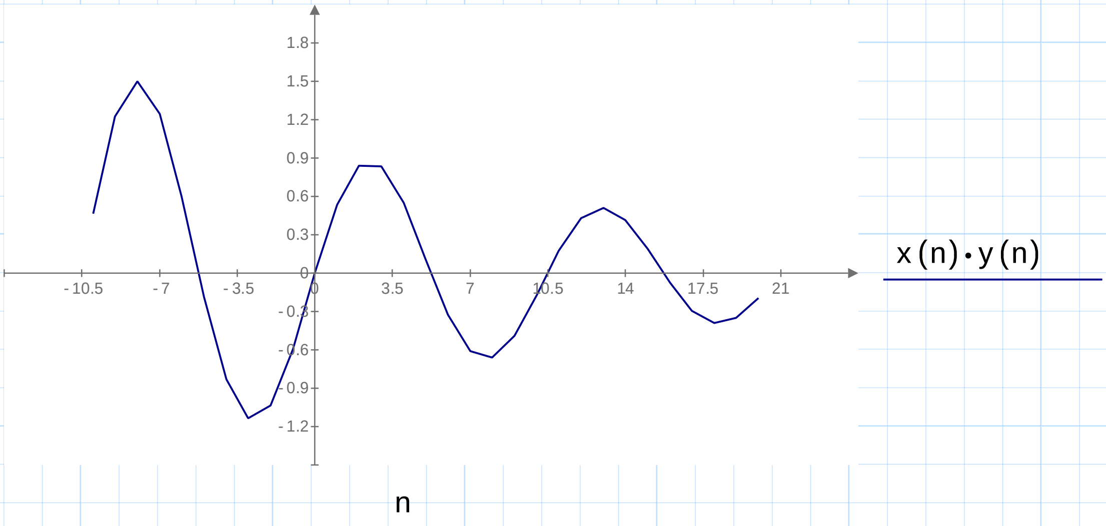
$x(t) := 3 \cdot \cos(2\pi \cdot t)$



<u>Example 2.4 continous signal</u>

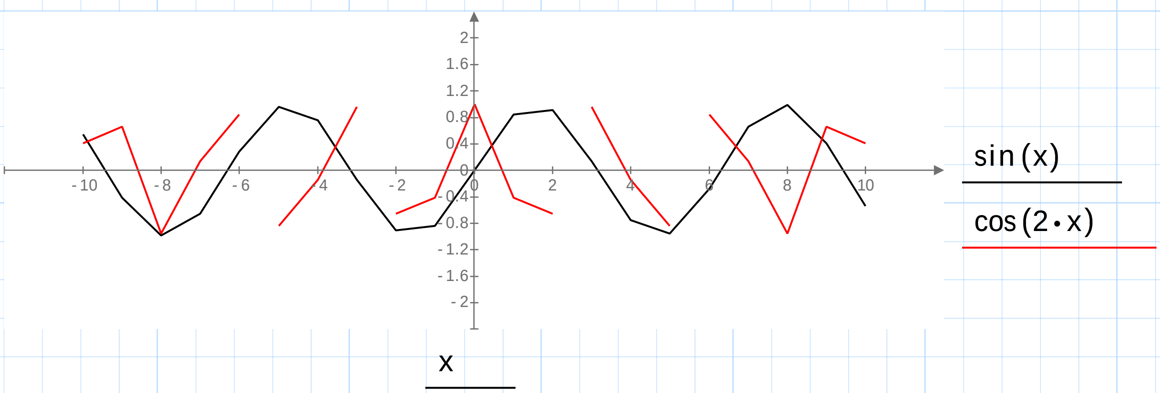
Examples 2.4 - 2.6 are aperiodic signals, ie they are not periodic signals

$t_{start} := -10$
$t_{end} := 20$
$n := t_{start} .. t_{end}$
$\alpha := 0.95 \qquad \omega := 0.6$
$y(n) := \alpha^n \qquad x(n) := \sin(\omega \cdot n)$



Decaying sine wave above plot.

<u>Example 2.5 continous signal</u>
$x := -10 .. 10$

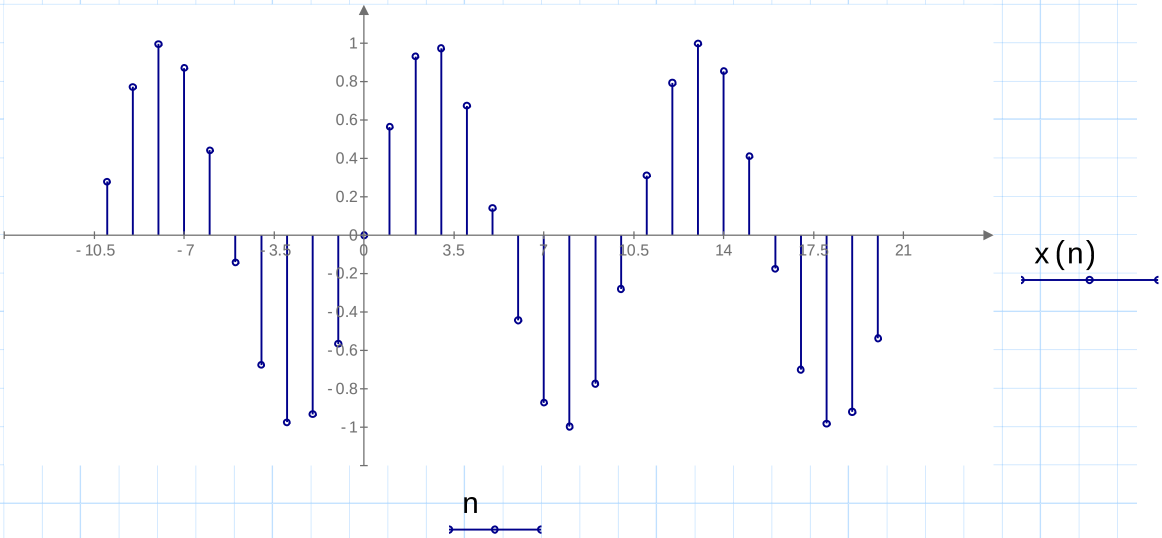


The signals are continous in time but discontinous in amplitude for cos(2x).

Example 2.6 discrete signal

$t_{start} := -10$
$t_{end} := 20$
$n := t_{start} .. t_{end}$
$\omega := 0.6$
$x(n) := \sin(\omega \cdot n)$

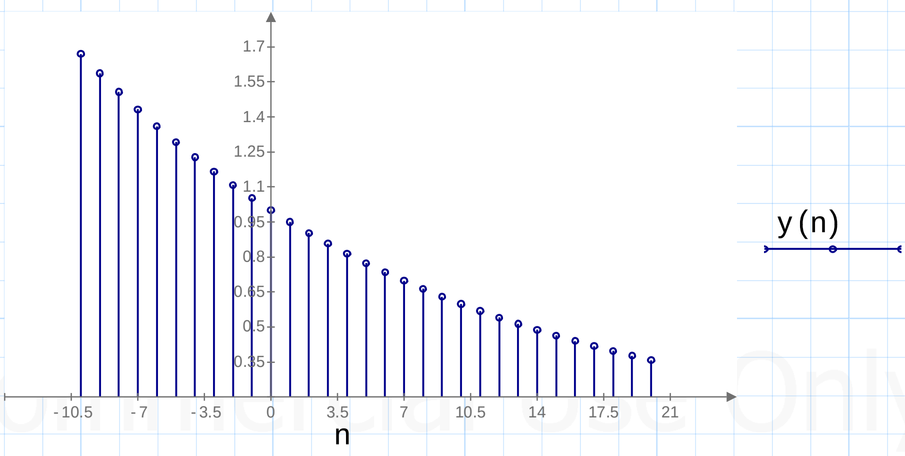
Set plot to type stem trace in Trace Group/Panel



In these exercies discrete functions are represented as x(n)

Example 2.7 discrete signal (exponential signal)

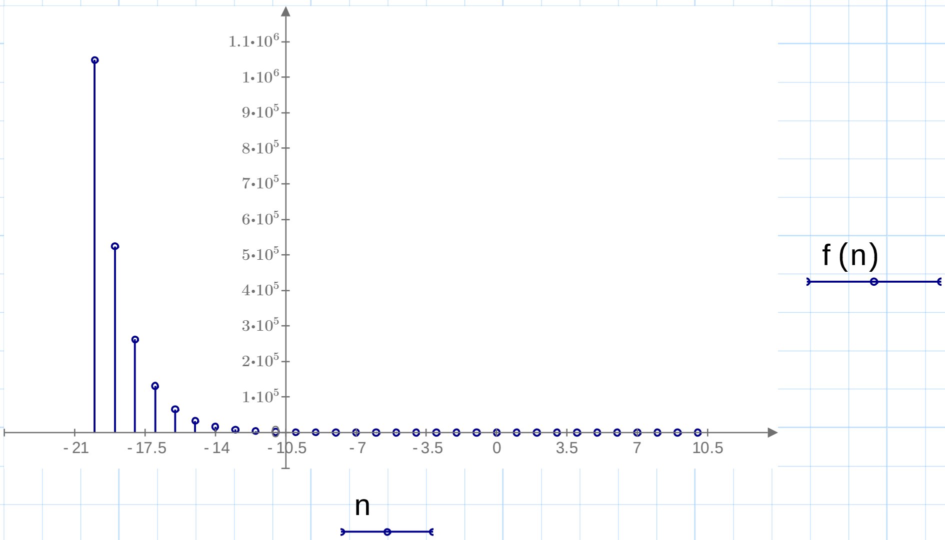
$t_{start} := -10$
$t_{end} := 20$
$n := t_{start} .. t_{end}$
$\alpha := 0.95$
$y(n) := \alpha^n$



Example 2.8a: Rule for calculating the nth value of the sequence.

$t_{start} := 10$
$t_{end} := -20$
$n := t_{start} .. t_{end}$

$$f(n) := \left\| \begin{array}{l} \left(\frac{1}{2}\right)^{n} \\ \text{if } n \geq 0 \\ \quad \| \, 0 \end{array} \right.$$



Values for n= 10 and -20 shown below. n>=0 range is very close to zero which is set to zero in the if else statement ie last line.

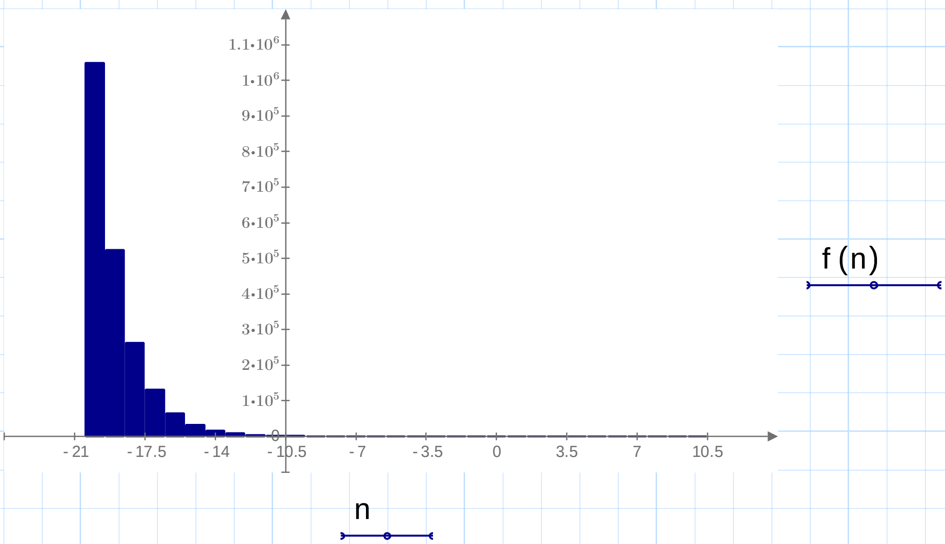
$$f\_10 := \left(\frac{1}{2}\right)^{10} = 0.000977$$

$$f\_minus\_20 := \left(\frac{1}{2}\right)^{-20} = 1.049 \cdot 10^{6}$$

Example 2.8b: column plot.

$t_{start} := 10$
$t_{end} := -20$
$n := t_{start} .. t_{end}$

$$f(n) := \left\| \begin{array}{l} \left(\frac{1}{2}\right)^{n} \\ \text{if } n \geq 0 \\ \quad \| \; 0 \end{array} \right.$$



Values for n= 10 and -20 shown below. n>=0 range is very close to zero which is set to zero in the if else statement ie last line.

$$f\_10 := \left(\frac{1}{2}\right)^{10} = 0.000977$$

$$f\_minus\_20 := \left(\frac{1}{2}\right)^{-20} = 1.049 \cdot 10^{6}$$

<u>Example 2.9 create a periodic sawtooth wave</u>

Demonstration of a recursive funtion to calculate factorial:

$factorial(n) := \mathbf{if}(n = 0, 1, n \cdot factorial(n-1))$

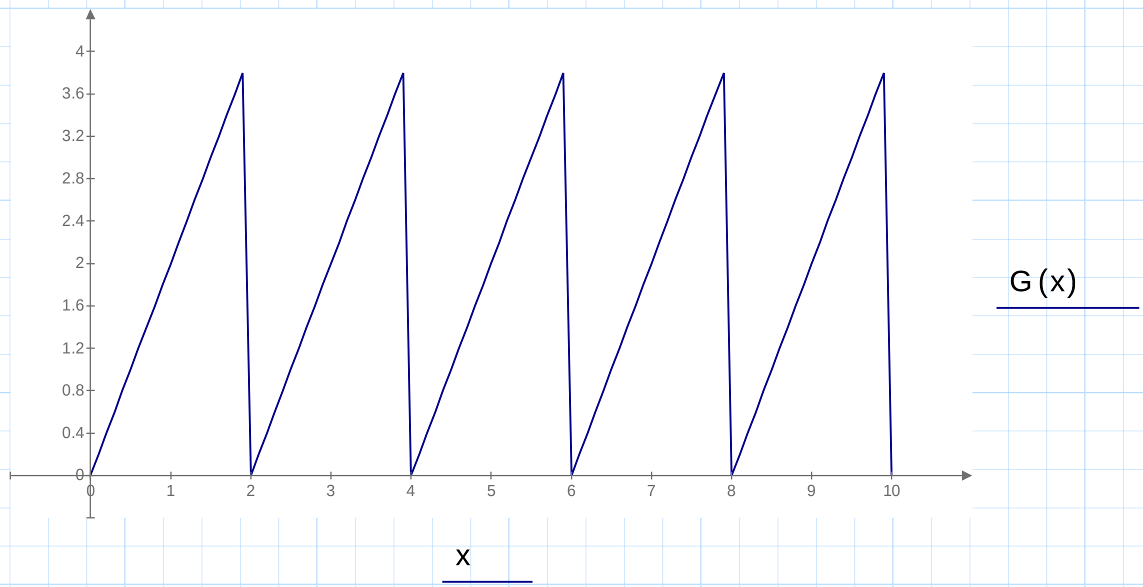
$factorial(1) = 1$
$factorial(5) = 120$

Now we use recursion to define the period of the sawtooth plot

$F(x) := 2 \cdot x$
$period := 2$

$G(x) := \mathbf{if}(x < period, F(x), G(x - period))$

$x := 0, 0.1 .. 10$

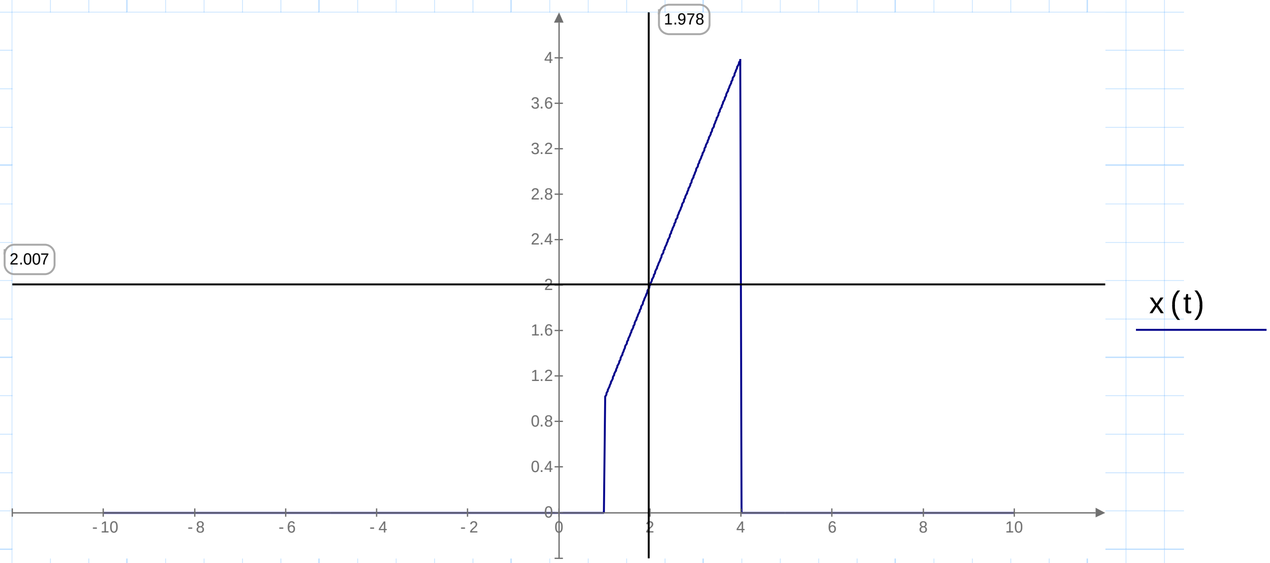


<u>Example 2.10 using function mod which returns the remainder of n when divided by k, to create periodic signals</u>

$u(t) := \Phi(t)$ Heaviside function enter F Ctrl G, capital letter F not f.

$t := -10, -9.99 .. 10$

$x(t) := t \cdot u(t-1) - t \cdot u(t-4)$ Defining the basis signal



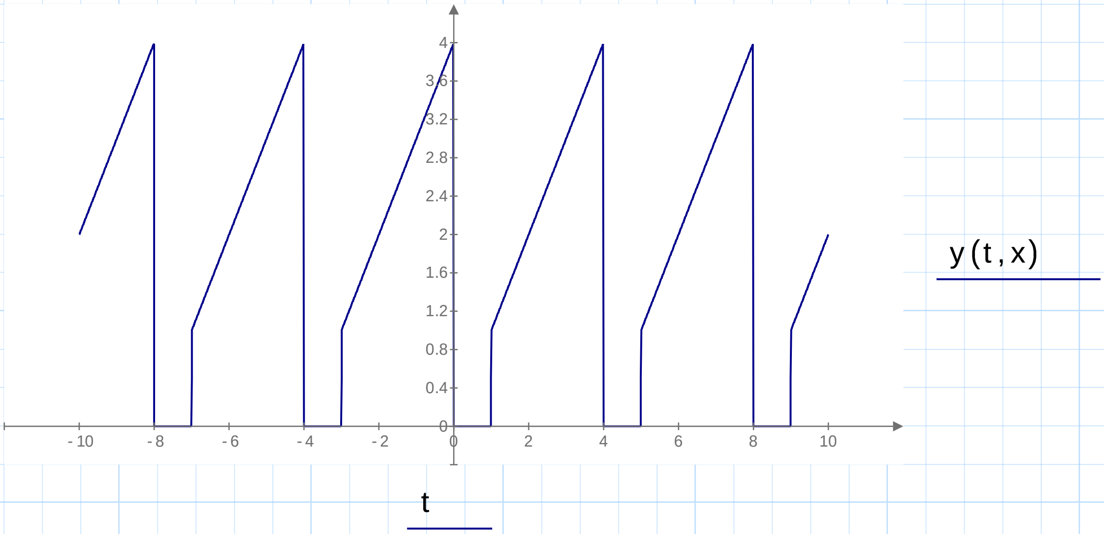
Heaviside funtion equal 1 for positive values.

at t = 2
x(t) = t u(t-1)-t u((t-4)   so look at t = 1 to t=4, like this -->u(t-4)-->t-4=0 t=4
2(u(2-1))-2(u(2-4))
2(u(1))-2(u(-2))
2u(1) -2u(-2)
2(1) - 2(0)
2.
Plot above shows at t = 2, x(2) = 2.

Next

$m := 4$ defining the period

$y(t, u) := \textbf{if}(\text{mod}(t, m) < 0, u(\text{mod}(t, m) + m), u(\text{mod}(t, m)))$ Defining the periodic from the basis

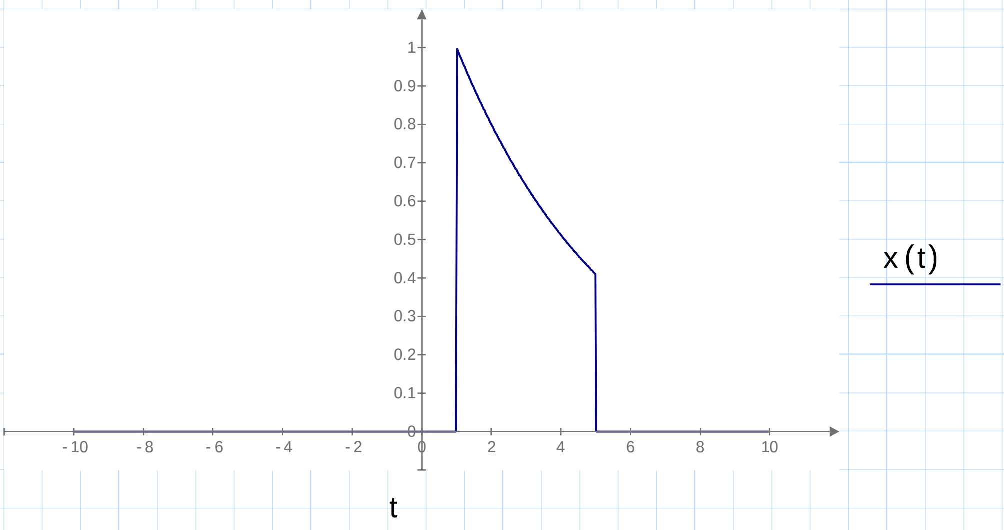


Next

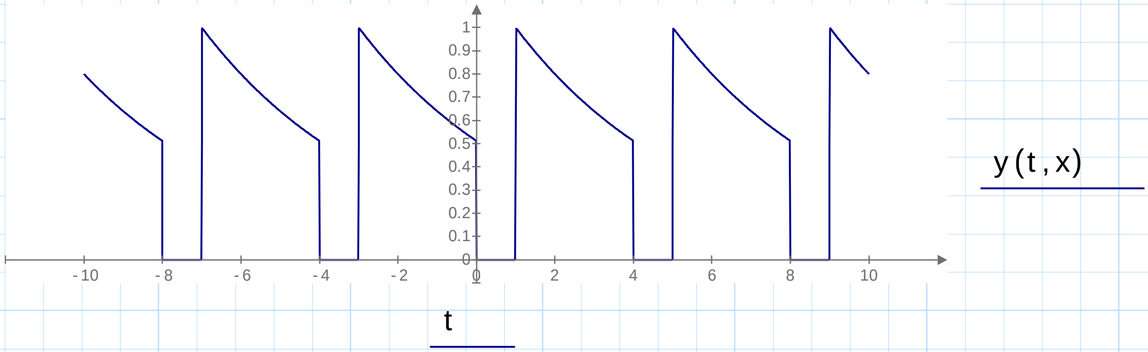
$m := 4$ the period

$y(t,u) := \textbf{if}(\text{mod}(t,m) < 0, u(\text{mod}(t,m) + m), u(\text{mod}(t,m)))$ The periodic function

$x(t) := \textbf{if}\left((1 \le t \le 5), (0.8)^{t-1} \cdot u(t-1), (0)\right)$



The curve in the plot above is set by x(t)



$t := -15, -14.99 .. 77$

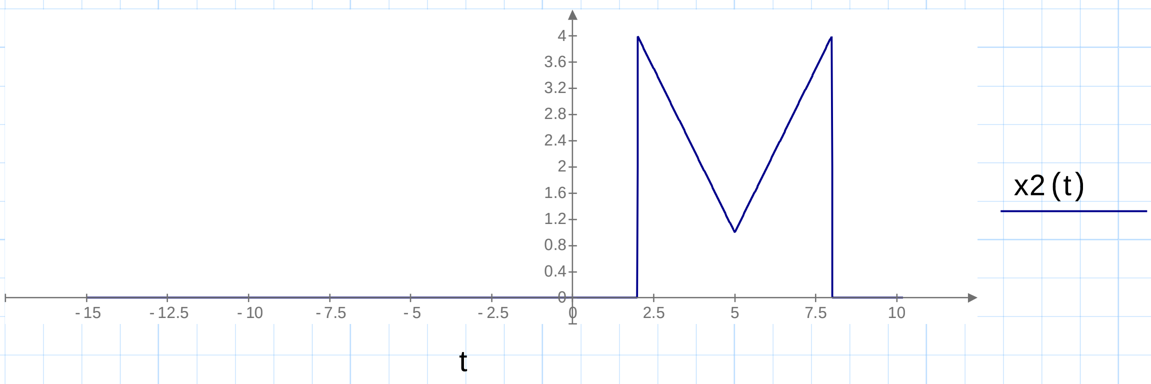
$x1(t) := t \cdot u(t-1) - t \cdot u(t-4)$ signal from previous example

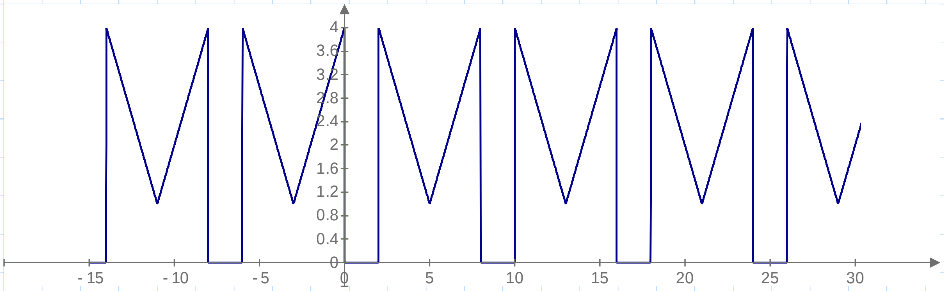
$x2(t) := x1(t-4) + (x1(-t+6))$ defining the new signal from the previous, x1(t) is in x2(t), first returning to x1(t) and then back to x2(t) for the final value

Now set m=8

$m := 8$ defining the period

$y(t, u) := \textbf{if}((\text{mod}(t, m) < 0), u(\text{mod}(t, m) + m), (u(\text{mod}(t, m))))$ defining the periodic signal

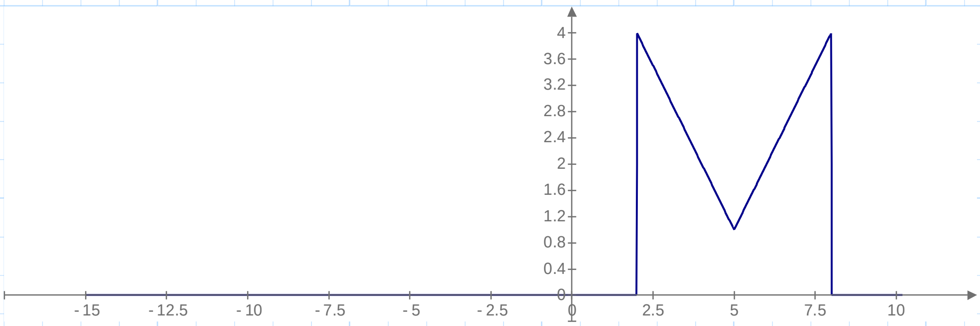




Next

$m := 7$

$y(t,u) := \mathbf{if}((\text{mod}(t,m) < 0), u(\text{mod}(t,m)+m), (u(\text{mod}(t,m))))$ defining the function

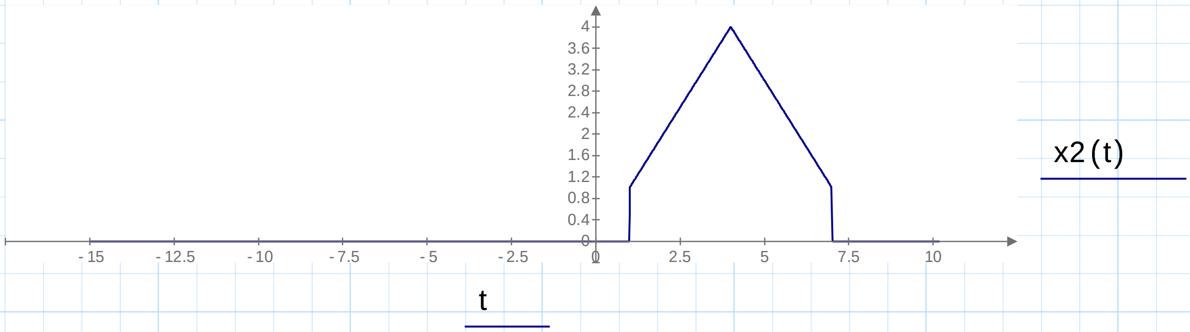


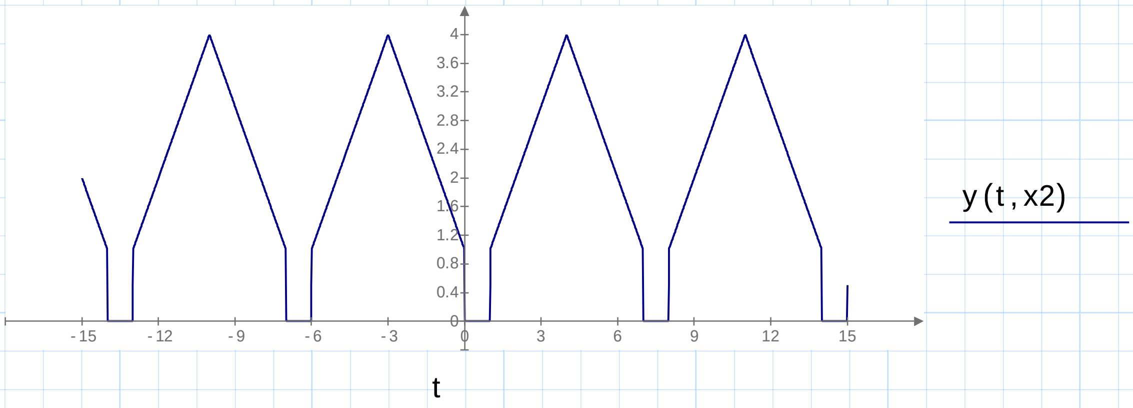
$u(t) := \Phi(t)$ defining the step function (F Ctrl G)

$t := -15, -14.99 .. 15$ set the range for t

$x(t) := t \cdot u(t-1) - t \cdot u(t-4)$ defining the basis signal

$x2(t) := x(t) + x(-t+8)$ shift the basis signal to appropriate unit





Re-initialise Variables:
Here we are going to set the variables to zero so that there are no errors.

x := 0   y := 0   x (t) := 0   y (t) := 0   y (x , t) := 0   x1 (t) := 0   x2 (t) := 0

y (x1 , t) := 0   y (x2 , t) := 0   n := 0

Example 2.11 White Noise (Noise Generators)

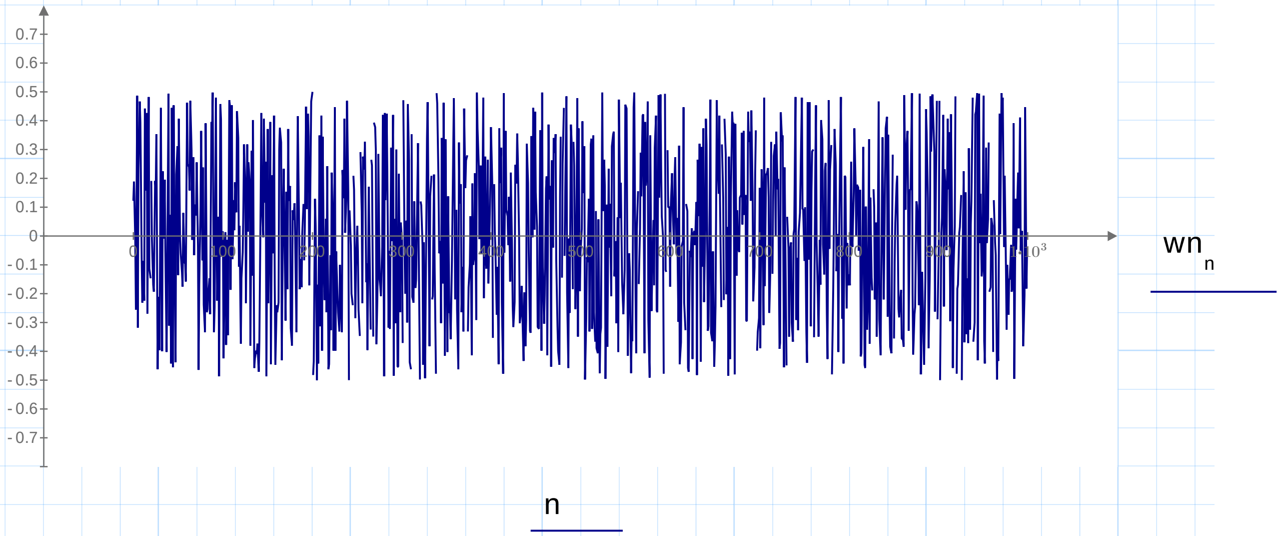
White and colored noise gets their names from the amplitude spectrum of the noise.

White noise demonstrates equal energy at all frequencies, in a similar manner to white color containing all colors at the same frequency.

Colored noise demonstrates more energy at some frequencies than others, just as colored light.

wn := whiten(1000) White noise function 'whiten'

n := 0 .. last(wn) entry 'wn_n' in the plot below has n as a matrix element n -vector



mean(wn) = 0.006777 close to zero, as seen from the plost above almost symmetrical to zero (pos and neg directions)

stdev(wn) = 0.288

$$\text{standarddev} := \sqrt{\left(\frac{1}{12}\right)} = 0.289$$

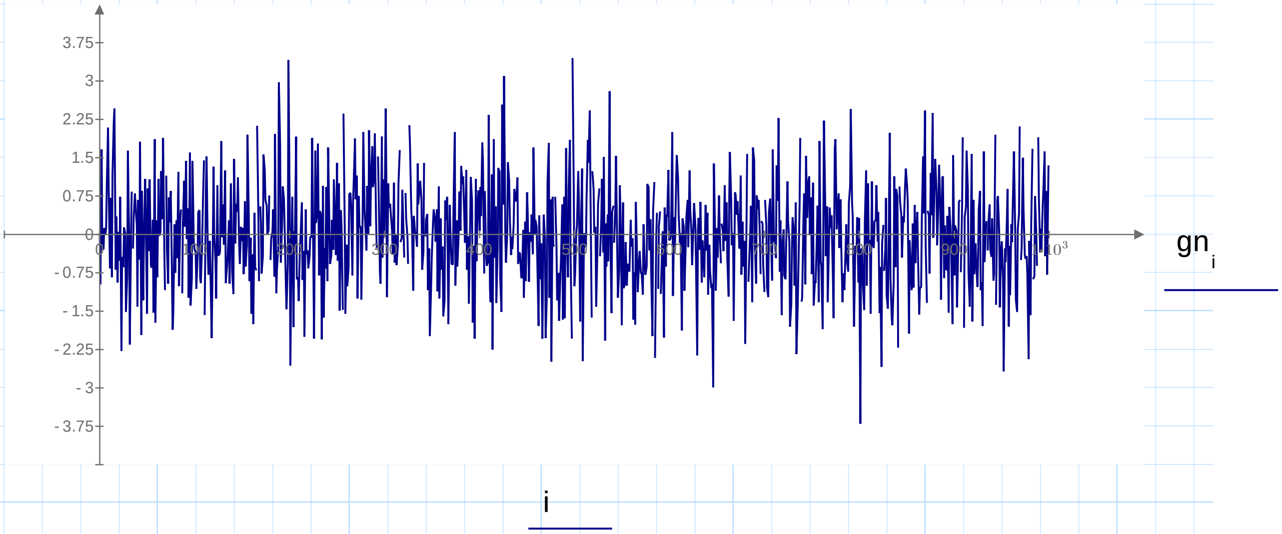
wn std dev is equal to sqrt of (1/12)

Note the mean and std dev of the white noise vector do not deviate much from the theorectical values.

Example 2.12 Gaussian Noise

The function gaussion (n) simulates n independent sources, each following a gaussian probability distribution of mean 0, and standard deviation 1. This type of noise is present any time we sum a large number of independent voltages.

gn := gaussn(1000)
i := 0 .. last(gn)



mean (gn) = 0.000075 similar to white noise

stdev (gn) = 1.019

Example 2.13 Gaussion noise in bell shape plot

Plot gaussion data in its more familiar bell-shaped form:

Number of bins 20
$nslots := 20$
$j := 0 .. (nslots - 1)$

Following vector holds the edges of the bins:

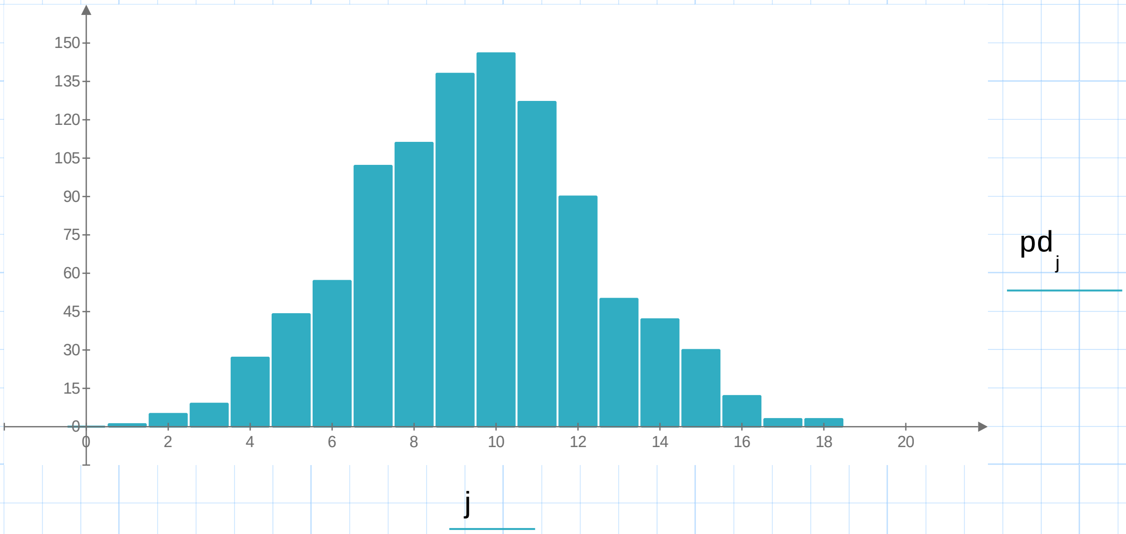
$slots_j := (-3.5) + j \cdot \left(\frac{7}{nslots}\right)$ change the number 7 in the numerator for a different plot, it will have the bell curve, also changing -3.5 shifts the curve

Set the rightmost boundary:

$slots_{nslots} := 3.5$

Now we fill the histogram:

pd := hist (slots , gn) function histogram 'hist'
set plot type to column trace



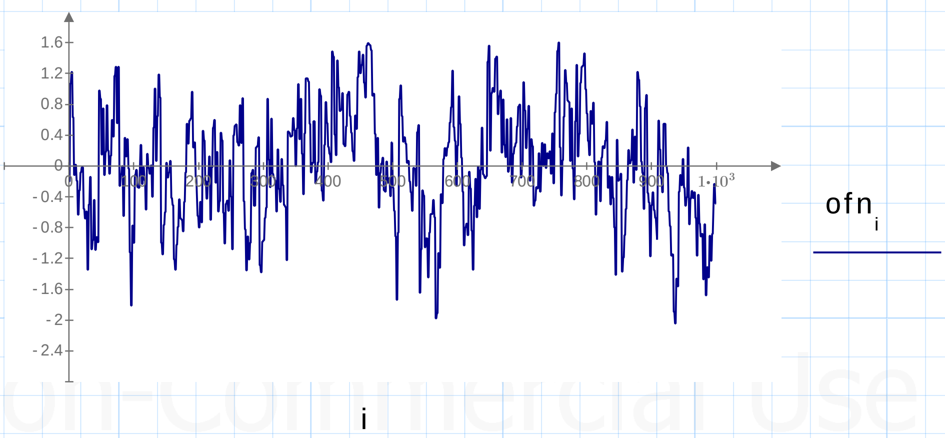
mean (pd) = 52.474

stdev (pd) = 49.388

Example 2.14 1/f noise:

Special type of white noise, also called pink noise. It is used to model a number of real world processes, such as audio signal processing. Pink noise also describes the instability of many common oscillators.

ofn := onefn (1000) function pink noise (1/f) is 'onefn'
ofn := ofn − mean (ofn) this line shifts the signal ofn by mean (ofn)
i := 0 .. last (ofn)



Example 2.15 Test input signal: Unit Step Function

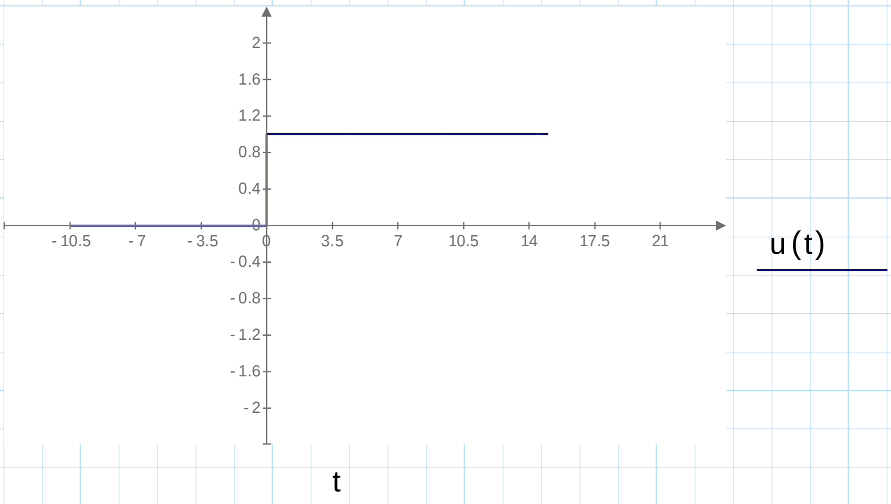
Two of the most common and useful test input signals are:
a. unit step function u(t)
b. unit impulse function d(t)

Unit step:

$t_{start} := -10$
$t_{end} := 20$
$n := t_{start} .. t_{end}$

$$u(n) := \begin{array}{l} \text{if } n \geq 0 \\ \quad u \leftarrow 1 \\ \text{else if } n \leq 0 \\ \quad u \leftarrow 0 \end{array}$$

Use variable t in the plot



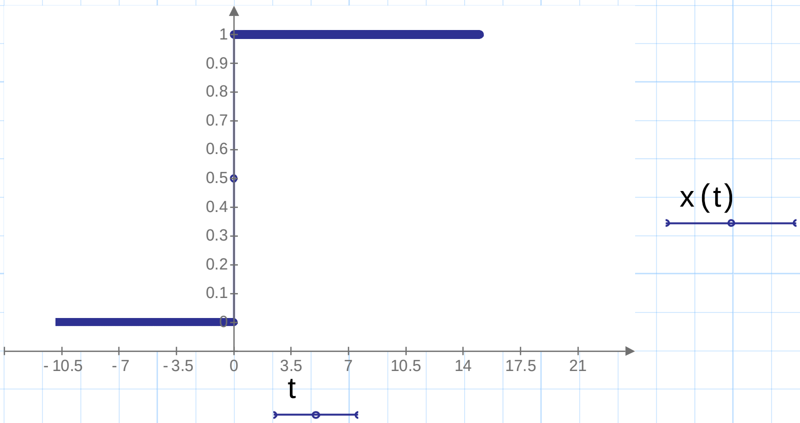
Using the Heaviside function for creating the Unit Step function:

$t_{start} := -10$

$t_{end} := 20$

$n := t_{start} .. t_{end}$ Use variable t in the plot

$x(n) := \Phi(n)$ Heaviside function enter F Ctrl G, capital letter F not f.



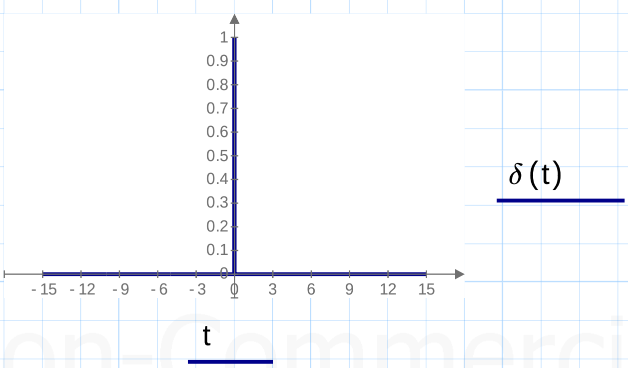
Example 2.16 Unit Impulse function

$t_{start} := -10$

$t_{end} := 20$

$n := t_{start} .. t_{end}$

$$\delta(t) := \begin{cases} \text{if } t = 0 \\ \quad \delta \leftarrow 1 \\ \text{else} \\ \quad 0 \end{cases}$$



Example 2.17 Shifting for t0 = 2

To shift u(t) or d(t) by a unit fo t0, replace t by t-t0 so u(t-t0) or d(t-t0)

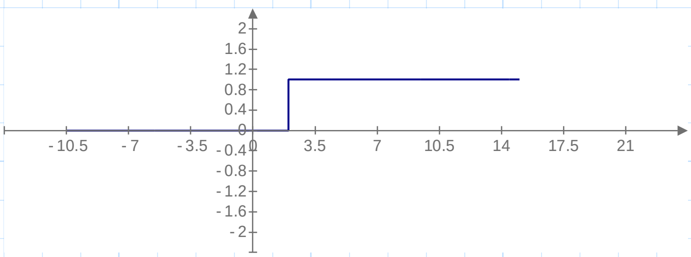
For scaling multiply the function by the scalar.

Unit step shift:

$t_{start} := -10$
$t_{end} := 20$
$n := t_{start} .. t_{end}$

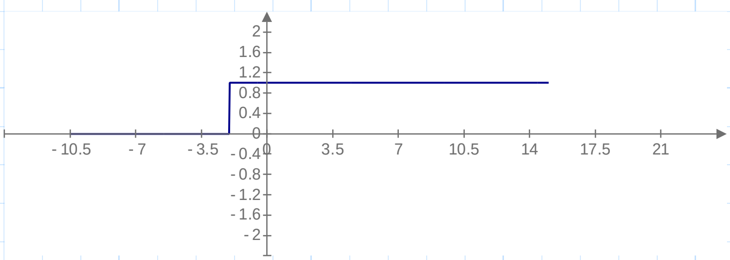
$$u(n) := \begin{cases} \text{if } n \geq 0 \\ \quad u \leftarrow 1 \\ \text{else if } n \leq 0 \\ \quad u \leftarrow 0 \end{cases}$$

Use variable t in the plot
Notice the shift in the plot where u(t) is defiend as u(t-2)
Move t to the right by 2.
t-2=0, t = 2 units to the right
to the left t+2 shown below.



$u(t-2)$

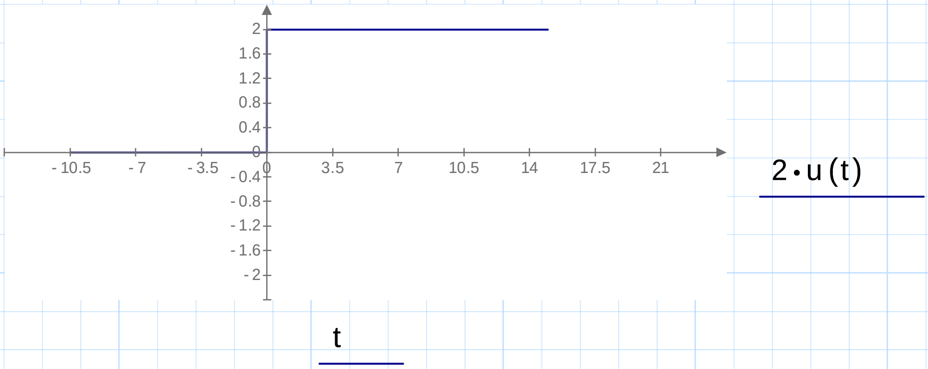
t



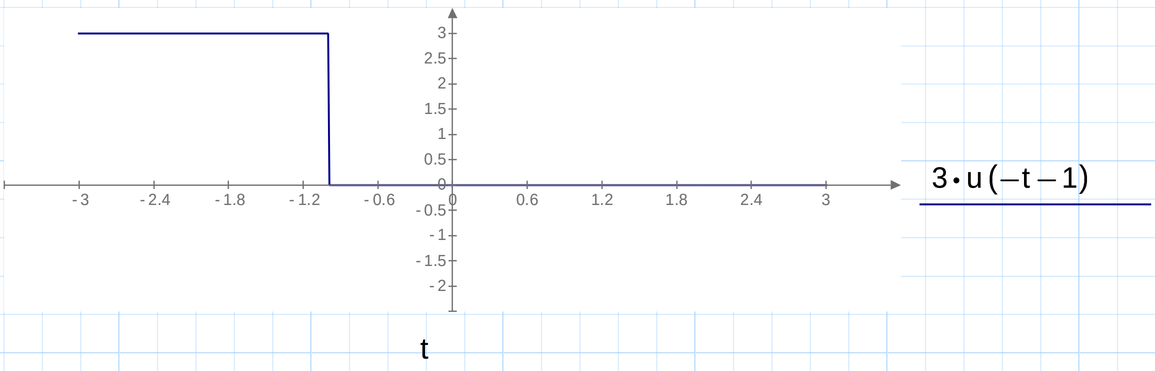
$u(t+2)$

t

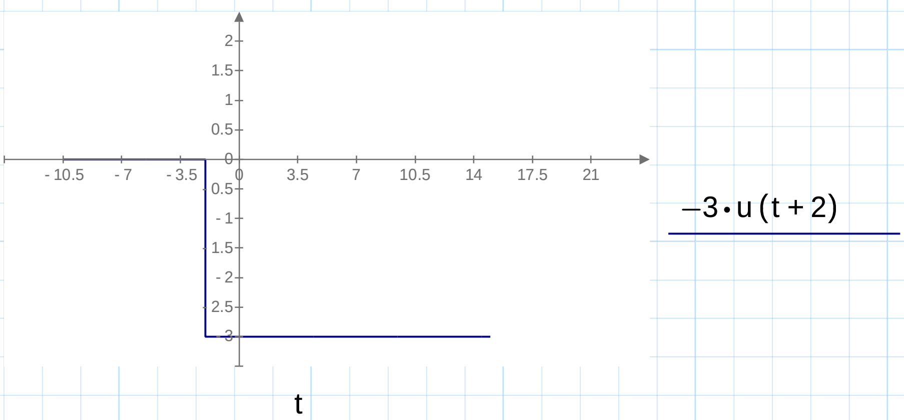
Now with scaling a = 2, and t = 2. So a x u(t)



Scaling for a = 3, t0 = 1, and t = -t:



Scaling for a = 3, t0 = - 2, and t = -t



Similary now for the unit impulse function:

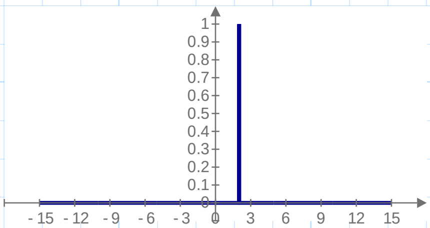
$t_{start} := -10$

$t_{end} := 20$

$n := t_{start} .. t_{end}$

$$\delta(t) := \begin{Vmatrix} \text{if } t = 0 \\ \quad \| \delta \leftarrow 1 \\ \text{else} \\ \quad \| 0 \end{Vmatrix}$$

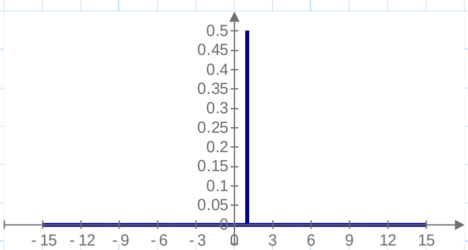
Shift for t0 = 2



$\delta(t-2)$

t

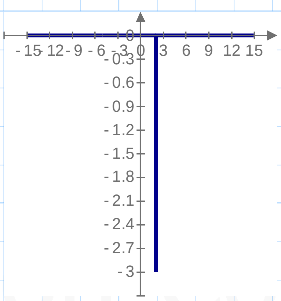
Scaling for a = 0.5, t0 = 1:



$0.5 \cdot \delta(t-1)$

t

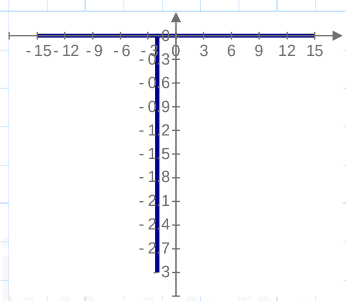
Scaling a = -3, t0 = 2 and t = t (positive)



$-3 \cdot \delta(t-2)$

t

Scaling a=-3, t0=2, t=-t (negative)
-3 delta(-t-2)



$-3 \cdot \delta(-t-2)$

t

Example 2.18 Even and Odd signals (PLUS Notes).

A signal x(t) or x(n) is called even if x(t)=x(-t)
or if x[-n]=x[n].

A signal x(t) or x(n) is called odd if x(-t)=-x(t)
or if x[-n]=-x[n].

We can represent any signal as a sum of even and odd funcions.

Where the even part xe(t) = (1/2) (x(t) + x(-t))
Where the odd part xo(t) = (1/2) (x(t) - x(-t))

Mathcad manipulates operations algebraically, so we use this feature
to express a signal as an even or odd sum.

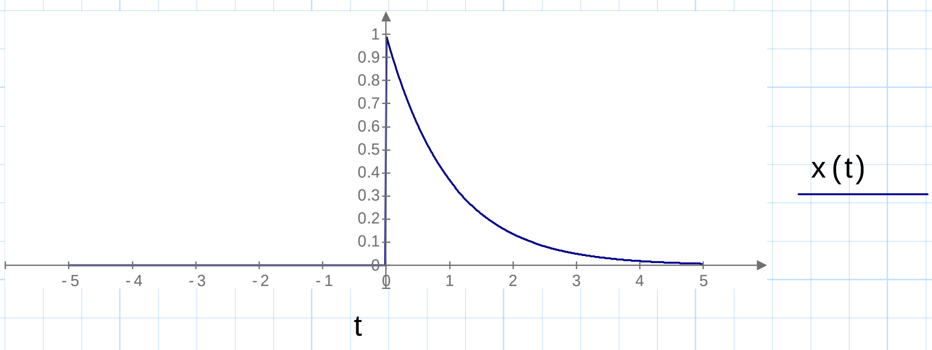
Example 2.18

$x(t) := e^{-t} \cdot u(t)$ this is the signal

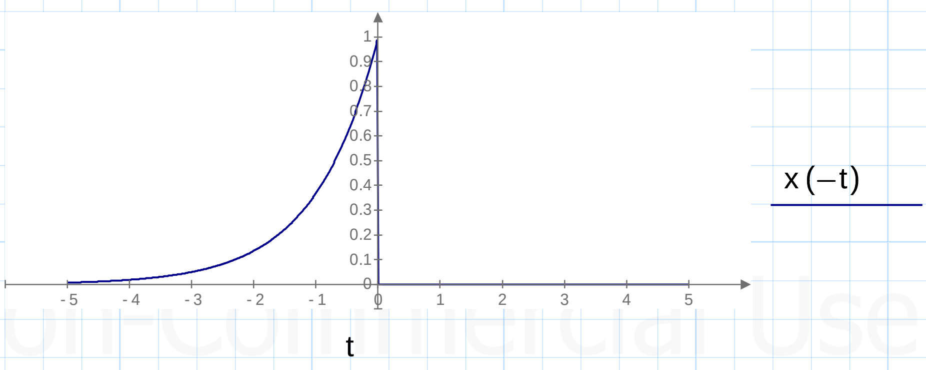
$t := -5, -4.99 .. 5$

$u(t) := \Phi(t)$ defining the unit step function

$x(t) := e^{-t} \cdot u(t)$



Signal shifted by negative sign, -t.



Continuing with the even and odd signal:

$x(t) := e^{-t} \cdot u(t)$ this is the signal

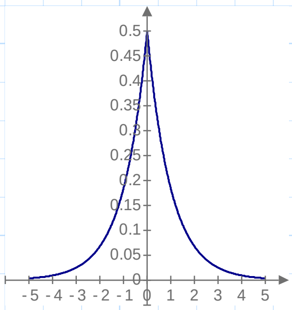
$t := -5, -4.99 .. 5$

$u(t) := \Phi(t)$ defining the unit step function

$x(t) := e^{-t} \cdot u(t)$

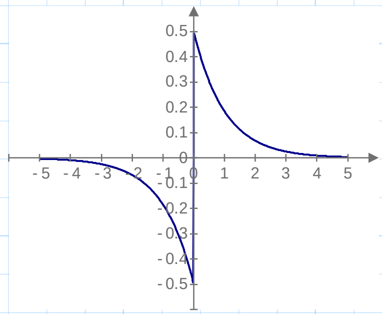
$$x_e(t) := \left(\frac{1}{2}\right) \cdot (x(t) + x(-t))$$

$$x_o(t) := \left(\frac{1}{2}\right) \cdot (x(t) - x(-t))$$



$x_e(t)$

t



$x_o(t)$

t

Plot of an even signal is symmetric about t =0, and plot of an odd signal is asymmetric.

An odd function will intergrate to zero over an integral t = -T to t = T.

Whereas an even signal will adhere to the following relationship:

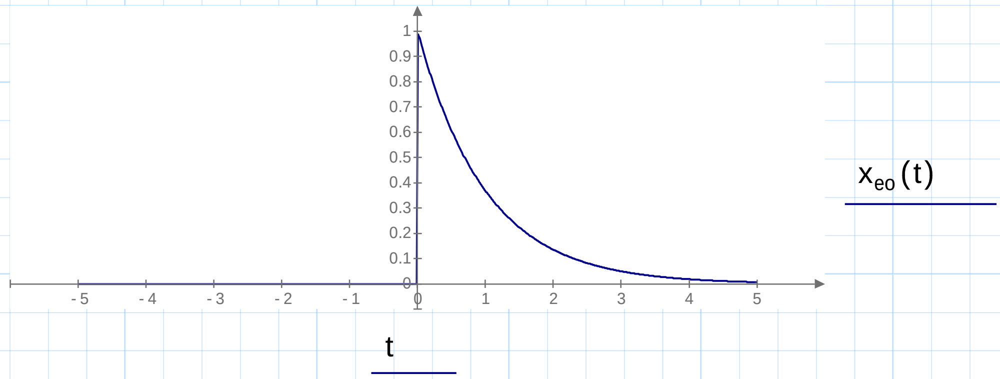
$$\int_{-T}^{T} x(t)dt = 2\int_{0}^{T} x(t)dt, \quad \text{for } x(t) \text{ even}$$

Product of even and odd signals rules:
1. (even)(even) = even
2. (even)(odd) = odd
3. (odd)(odd) = (even)

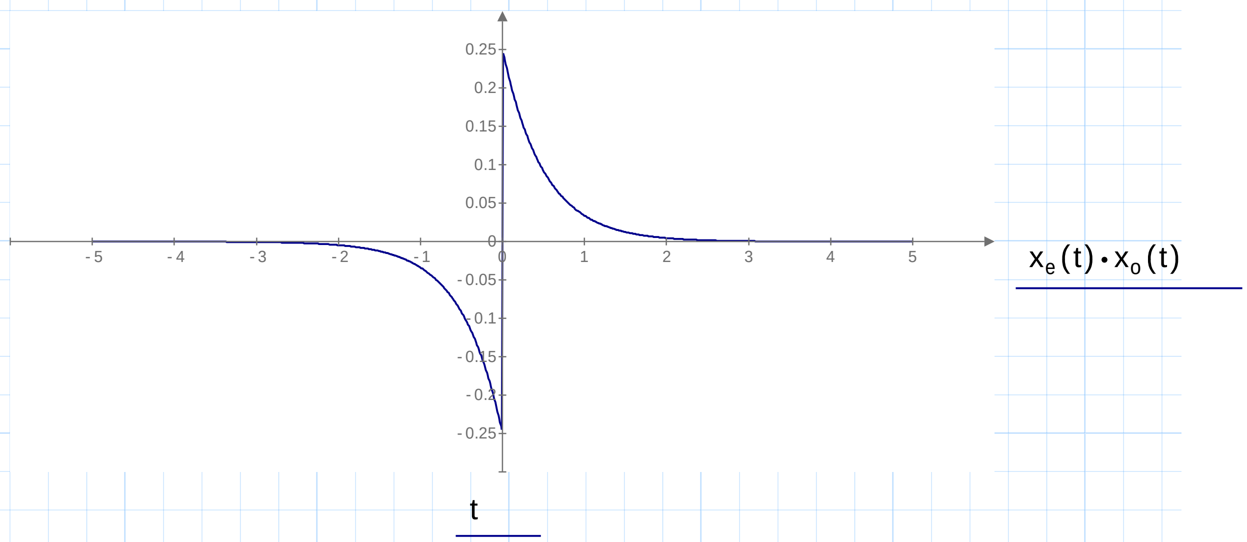
Orthogonal Signals:

2 signals are called orthogonal over a period of time t=t1 to t=t1+Tm, when their product integrates or averages to zero. Where m is an integer m=1,2,3.. Thus cos(wt) and sin(wt) are orthogonal for -T<t<T, and since these two signals are also periodic, (T0=2pi/w), they are also orthogonal within any interval from t=t1 to t=t1+mT0. Where m is any integer m=1,2,3..

$x_{eo}(t) := x_e(t) + x_o(t)$



From the previous even and odd signals, their sum here shows a value in the positive quadrant, elsewhere its zero.



Example 2.19 Additional Examples Using Mathcad:

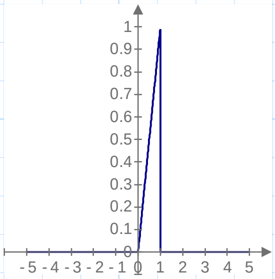
$u(t) := 0$

$t := -5, -4.99 .. 5$

$u(t) := \Phi(t)$ unit step function

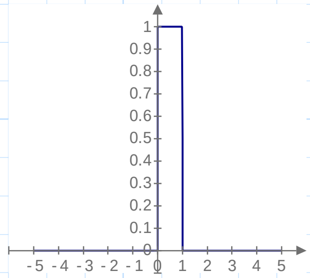
$x1(t) := t \cdot u(t) - t \cdot u(t-1)$ signal x1(t)

$x2(t) := u(t) - u(t-1)$ signal x2(t)



x1 (t)

t



x2 (t)

t

Signals after shifting and scaling:

x1(0.75t-1) scale 0.75 shift 0.75

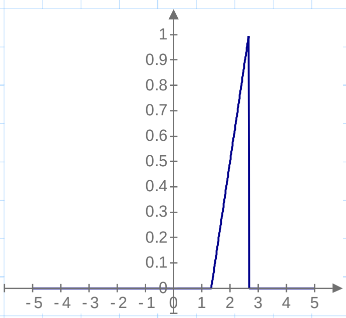
$a := 0.75$

$x11(t) := x1(a \cdot t - 1)$

x2(0.5t-0.5) scale 0.5 shift 0.5

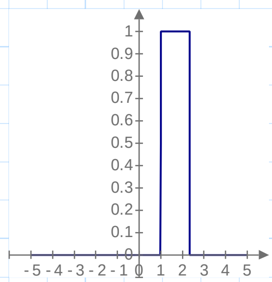
$b := 0.5$

$x22(t) := x2(a \cdot t - a)$



x11 (t)

t



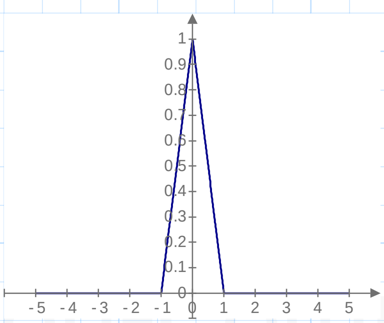
x22 (t)

t

Next we add and multiply the signals:

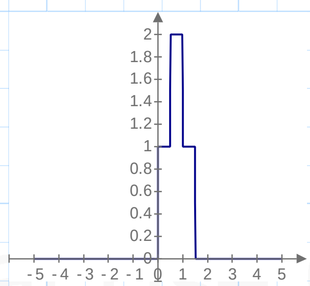
$x3(t) := x1(t+1) + x1(1-t)$

$x4(t) := x2(t) + x2(t-0.5)$



x3 (t)

t



x4 (t)

t