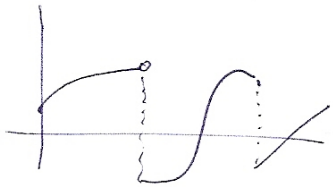
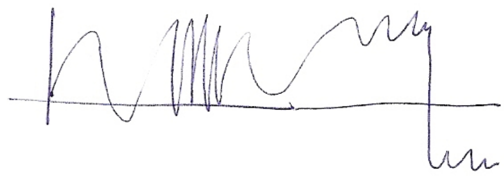
<u>Remark:</u> Discrete Fourier transform (DFT) of vector with N components requires about N^2 operations.
Fast Fourier transform (FFT) is efficient algorithm of DFT which requires only $\log_2 N \cdot N$ operations. E.g. if we take $N = 1024 = 2^{10}$ then DFT has $N^2 = 1048576$ op. and FFT $10 \cdot 1024 = 10240$, i.e. approx. 100 times less. <u>FFT</u> is also refered as <u>Cooley-Tukey</u> algorithm

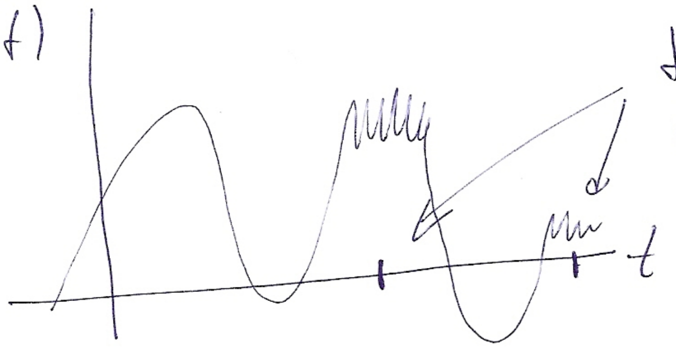
Time-frequency analysis of signal

→ classical Fourier transform is good for sufficiently smooth signal (function), since for smooth function Fourier coefficients (or amplitude spectra) goes fast to zero for increasing frequency, i.e. for discontinuous function (signal)  or

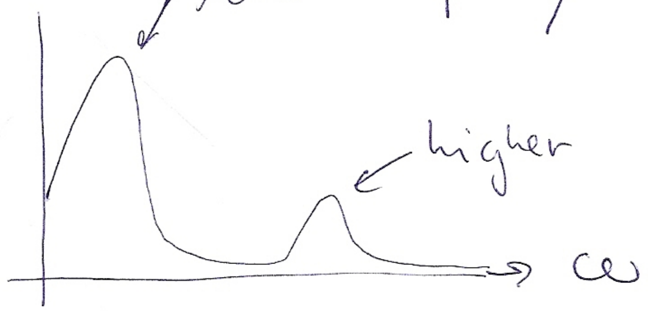
for "unsteady" signal 

we get "too many" non-zero coefficients (not good for real time signal transfer)

→ classical Fourier transform needs complete original function (signal) and provides amplitude spectra for whole signal, without any information about the appearence of particular frequencies, e.g.



the "position" of high frequency is lost



lover frequency

higher

$\Rightarrow$ motivation for "local" Fourier transform

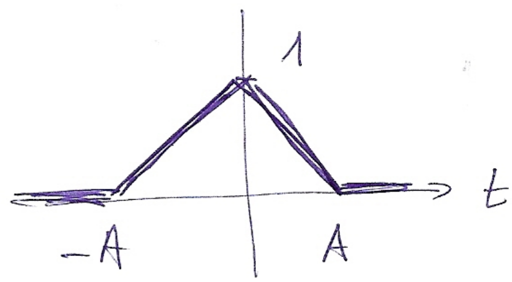
Windowed Fourier transform (WFT)

Consider we have signal $f(t)$, $t \in \mathbb{R}$.
We define the so called window $w(t)$ as

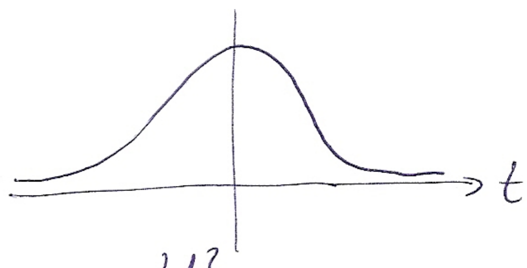
$$w(t) \begin{cases} \neq 0, & t \in (-A, A) \\ = 0, & \text{elsewhere} \end{cases}$$

Examples of windows:

- triangular window



- Gauss window



$$w(t) = A e^{-\lambda t^2}$$

The windowed Fourier transform is defined

$$\boxed{F(\omega, b) = \int_{-\infty}^{\infty} f(t)\, \overline{w}(t-b)\, e^{-i\omega t}\, dt}$$

It is "local" Fourier transform in the neighbourhood of $t = b$.

One of the first methods was

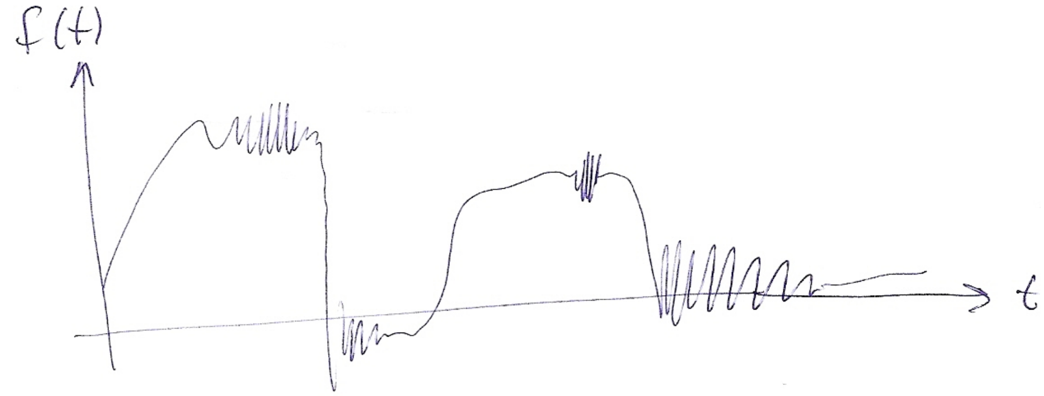
Gabor method (1946):

$$w(t) = \pi^{-1/4} e^{-t^2/2}$$

$$F(\omega, b) = \int_{-\infty}^{\infty} f(t)\, \overline{w}(t-b) \cdot e^{-i\omega t}\, dt$$

→ The disadvantage of this method is the fixed length of window, it can be hardly used for analysis of sound, turbulence, ..., where big differences of time scales appear,

e.g.

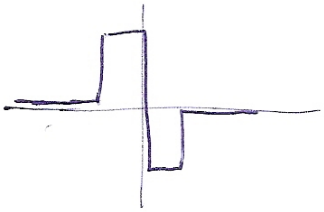
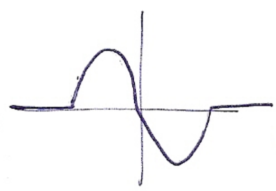


⟹ This fact motivated the development of method with variable window width (Morlet, Grossman 1984)

Wavelet analysis

The main idea is to write the signal $f(t)$ as a linear combination of functions $\psi_{ab}(t)$, where

$$\psi_{ab}(t) = \frac{1}{\sqrt{a}}\,\psi\left(\frac{t-b}{a}\right), \quad a, b \in \mathbb{R}, \quad a>0$$

and $\psi(t)$ is so called mother wavelet. We see, that a correspond to the period (i.e. $1/a$ is frequency) and b to the shift. Mother wavelet can be of different shape , , ...

The main advantage of wavelet analysis compared to WFT is, that the "support" gets smaller for higher frequencies (good for wide range of time scales in signal).

The wavelet transform is defined as

$$C_F(a,b) = \int_{-\infty}^{\infty} f(t) \cdot \overline{\Psi_{ab}(t)}\, dt$$

Rem: Good source of information can be found at (www.wavelet.org).

Example: Consider the signal

$$f(t) = \begin{cases} \sin(10t), & 0 \leq t \leq 2\pi \\ \sin(0{,}5t), & 2\pi \leq t \leq 10\pi \\ \sin(20t), & 10\pi \leq t \leq 11\pi \\ 0, & \text{elsewhere} \end{cases}$$

212

We will see

a) the classical Fourier transform as well as WFT of $f(t)$

b) some simple wavelet analysis of $f(t)$ with two different mother wavelets

213

```
> restart:
```

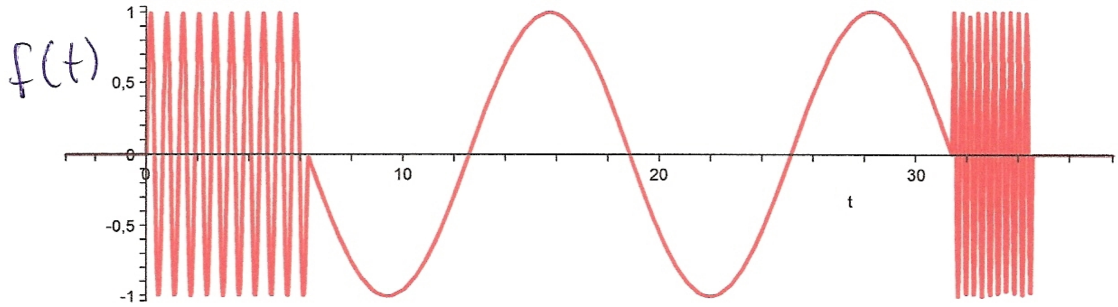
```
> with(plots):
```

Warning, the name changecoords has been redefined

```
> f(t):=piecewise(t<0,0,
                  t>=0 and t<2*Pi, sin(10*t),
                  t>=2*Pi and t<10*Pi, sin(0.5*t),
                  t>=10*Pi and t<11*Pi, sin(20*t),
                  t>=11*Pi, 0);
```

$$f(t) := \begin{cases} 0 & t < 0 \\ \sin(10\,t) & 0 \le t \textbf{ and } t < 2\,\pi \\ \sin(0.5\,t) & 2\,\pi \le t \textbf{ and } t < 10\,\pi \\ \sin(20\,t) & 10\,\pi \le t \textbf{ and } t < 11\,\pi \\ 0 & 11\,\pi \le t \end{cases}$$

```
> plot(f(t),t=-Pi..12*Pi);
```

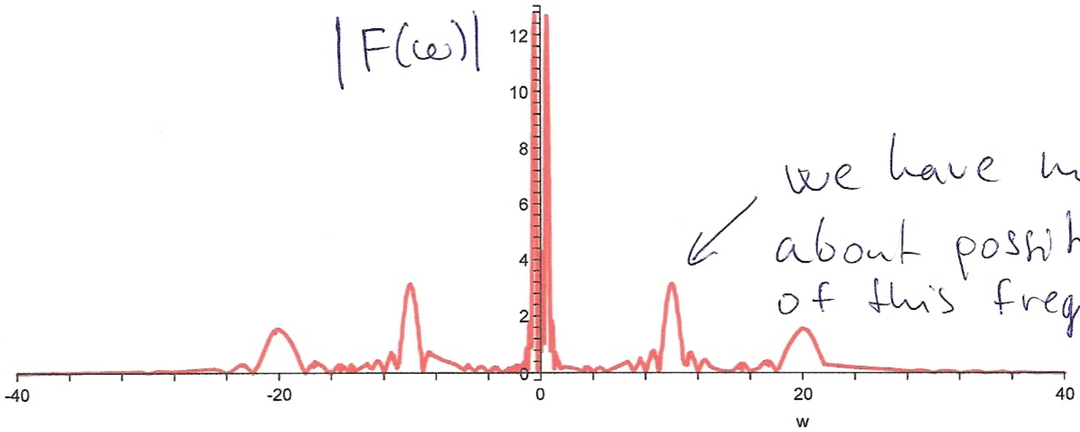



```
> F(w):=int(f(t)*exp(-I*w*t),t=0..11*Pi):
```

```
> F2(w):=sqrt(Re(F(w))^2+Im(F(w))^2):
```

```
> plot(F2(w),w=-40..40);
```

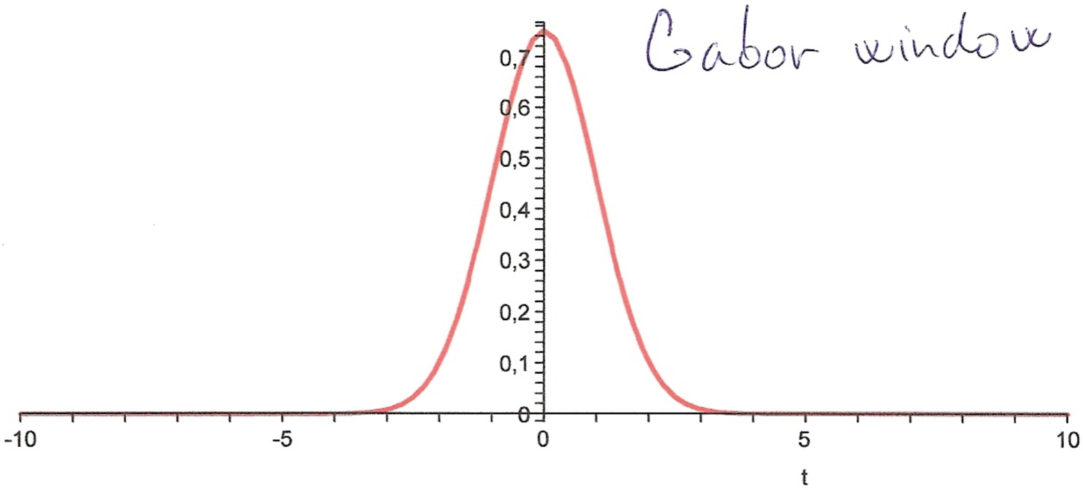
|F(ω)|



we have no info about position of this frequency

```
> g(t):=Pi^(-0.25)*exp(-t^2/2):
> plot(g(t),t=-10..10);
```

Gabor window

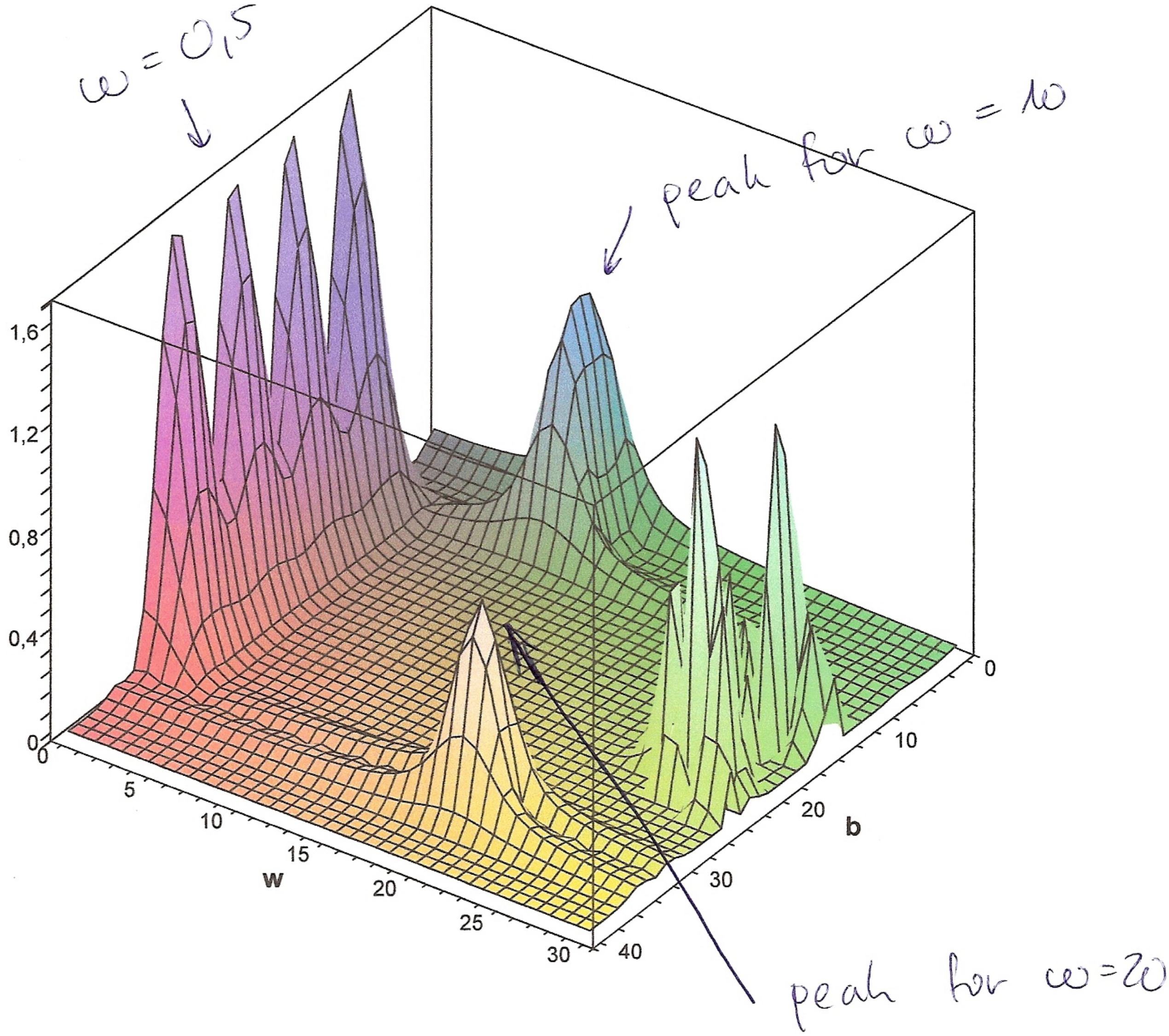


Gabor method

```
> FW(w,b):=int(f(t)*Pi^(-0.25)*exp(-(t-b)^2/2)*exp(-I*w*t),t=-2*Pi..13*Pi);
```

215

```
>
> plot3d(FW2(w,b),b= 0..40, w=0..30,
        style = patch,
        grid = [40,40],
        orientation = [5,25],
        axes=boxed,
        labelfont = [TIMES,BOLD,12]);
```

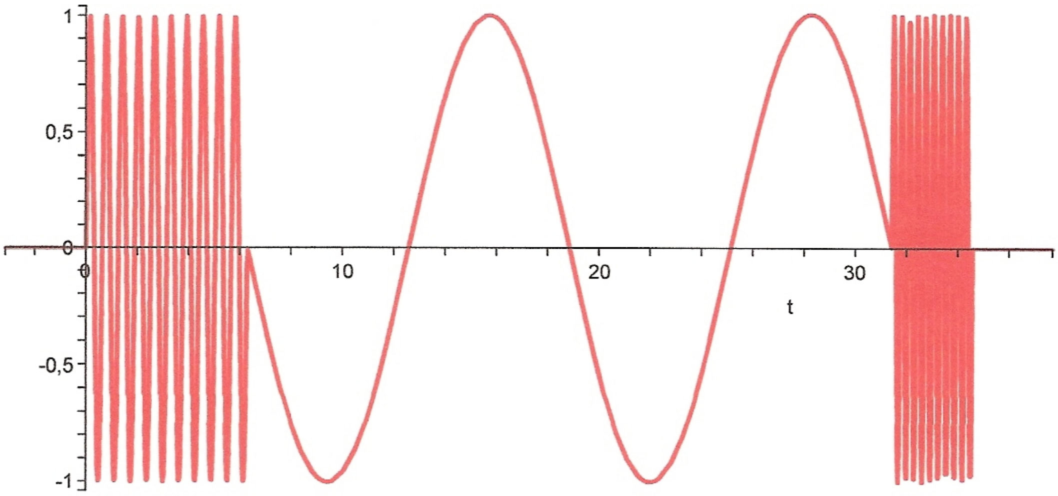


(216) 1st wavelet example

```
> restart:
> with(plots):
Warning, the name changecoords has been redefined
> f(t):=piecewise(t<0,0,
                  t>=0 and t<2*Pi, sin(10*t),
                  t>=2*Pi and t<10*Pi, sin(0.5*t),
                  t>=10*Pi and t<11*Pi, sin(20*t),
                  t>=11*Pi, 0);
```

$$f(t) := \begin{cases} 0 & t < 0 \\ \sin(10\, t) & 0 \le t \text{ and } t < 2\,\pi \\ \sin(0.5\, t) & 2\,\pi \le t \text{ and } t < 10\,\pi \\ \sin(20\, t) & 10\,\pi \le t \text{ and } t < 11\,\pi \\ 0 & 11\,\pi \le t \end{cases}$$

```
> plot(f(t),t=-Pi..12*Pi);
```



Note that instead of $\Psi_{ab}(t) = \frac{1}{\sqrt{a}}\Psi\left(\frac{t-b}{a}\right)$ from page 210 we use wavelet(t) =

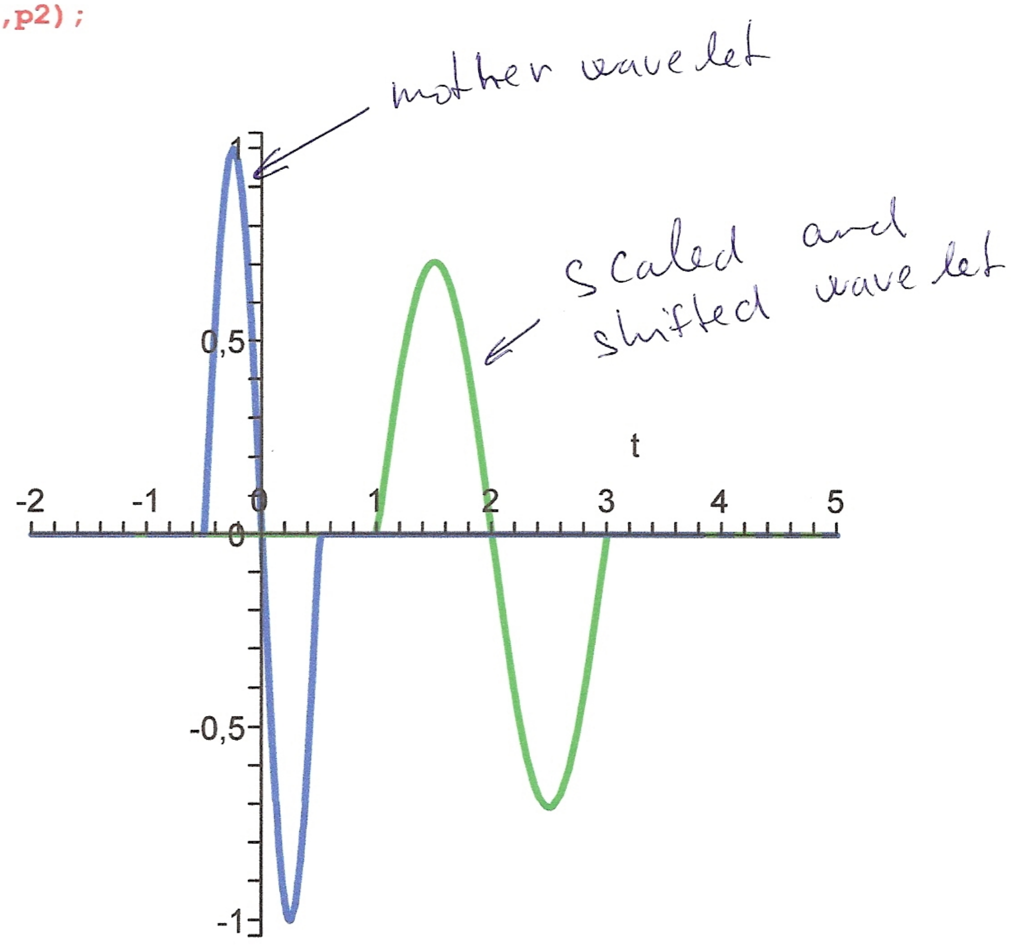
$$= \sqrt{\frac{\omega}{2\pi}} \cdot \Psi\left[(t-b)\cdot\frac{\omega}{2\pi}\right], \quad \text{where } \frac{\omega}{2\pi} = \frac{1}{a}$$

```
> wavelet(t):=piecewise((t-b)*w/2/Pi<-0.5,0,
                        (t-b)*w/2/Pi>=-0.5 and (t-b)*w/2/Pi<=0.5,
  -sqrt(w/2/Pi)*sin((t-b)*w),
                        (t-b)*w/2/Pi>0.5,0);
```

217

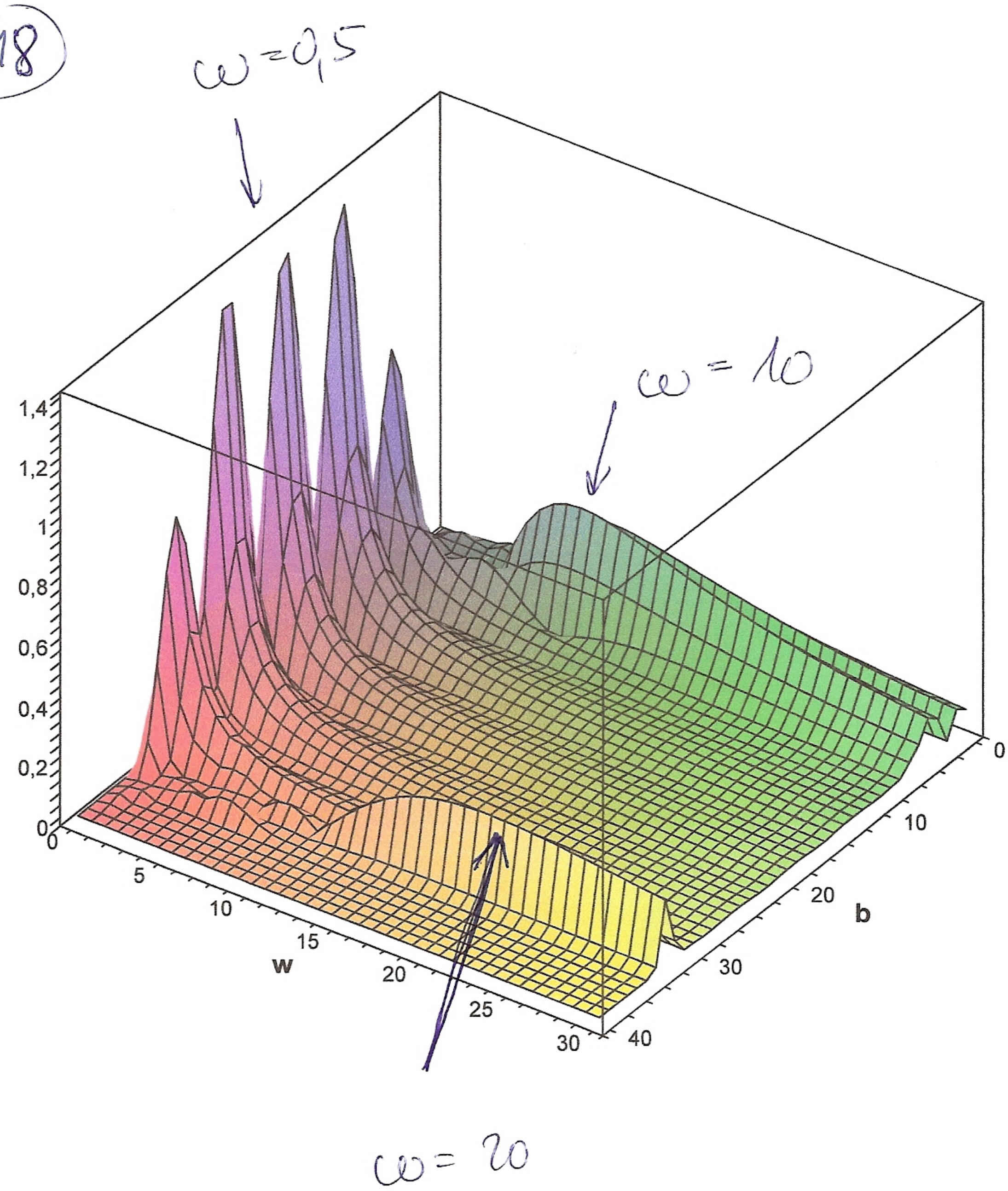
$$wavelet(t) := \begin{cases} 0 & \frac{(t-b)\,w}{2\,\pi} < -0.5 \\ -\frac{1}{2}\sqrt{2}\,\sqrt{\frac{w}{\pi}}\,\sin((t-b)\,w) & -0.5 \le \frac{(t-b)\,w}{2\,\pi} \text{ and } \frac{(t-b)\,w}{2\,\pi} \le 0.5 \\ 0 & 0.5 < \frac{(t-b)\,w}{2\,\pi} \end{cases}$$

```
> p1:=plot(subs(b=0,w=2*Pi,wavelet(t)),t=-2..5,color=blue):
> p2:=plot(subs(b=2,w=0.5*2*Pi,wavelet(t)),t=-2..5,color=green):
> display(p1,p2);
```




```
> FW(w,b):=int(f(t)*wavelet(t),t=-2*Pi..13*Pi):
> FW2(w,b):=sqrt(Re(FW(w,b))^2+Im(FW(w,b))^2):
> plot3d(FW2(w,b),b= 0..40, w=0..30,
        style = patch,
        grid = [40,40],
        orientation = [5,25],
        axes=boxed,
        labelfont = [TIMES,BOLD,12]);
```

218



>

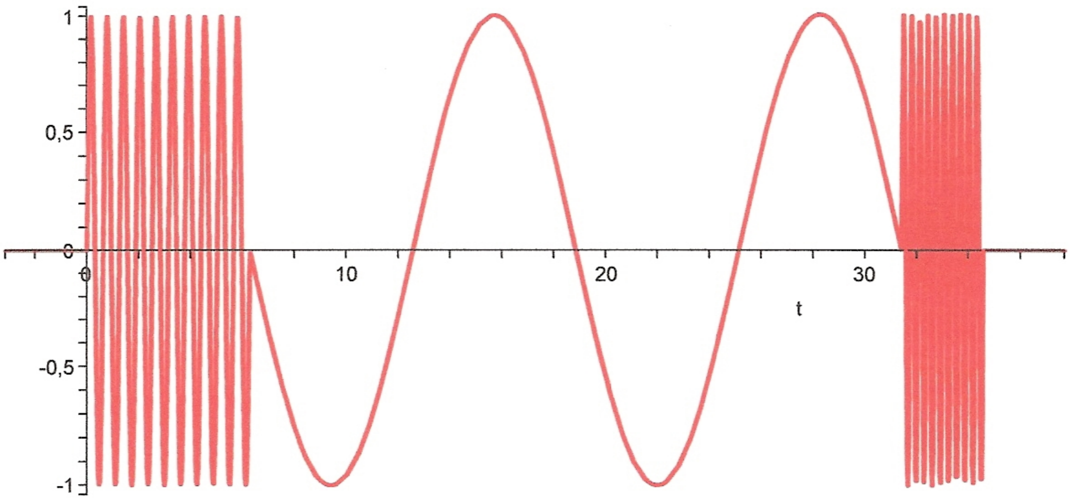
219

2nd example of wavelet

```
> restart:
> with(plots):
Warning, the name changecoords has been redefined
> f(t):=piecewise(t<0,0,
                 t>=0 and t<2*Pi, sin(10*t),
                 t>=2*Pi and t<10*Pi, sin(0.5*t),
                 t>=10*Pi and t<11*Pi, sin(20*t),
                 t>=11*Pi, 0);
```

$$f(t) := \begin{cases} 0 & t < 0 \\ \sin(10\,t) & 0 \leq t \textbf{ and } t < 2\,\pi \\ \sin(0.5\,t) & 2\,\pi \leq t \textbf{ and } t < 10\,\pi \\ \sin(20\,t) & 10\,\pi \leq t \textbf{ and } t < 11\,\pi \\ 0 & 11\,\pi \leq t \end{cases}$$

```
> plot(f(t),t=-Pi..12*Pi);
```



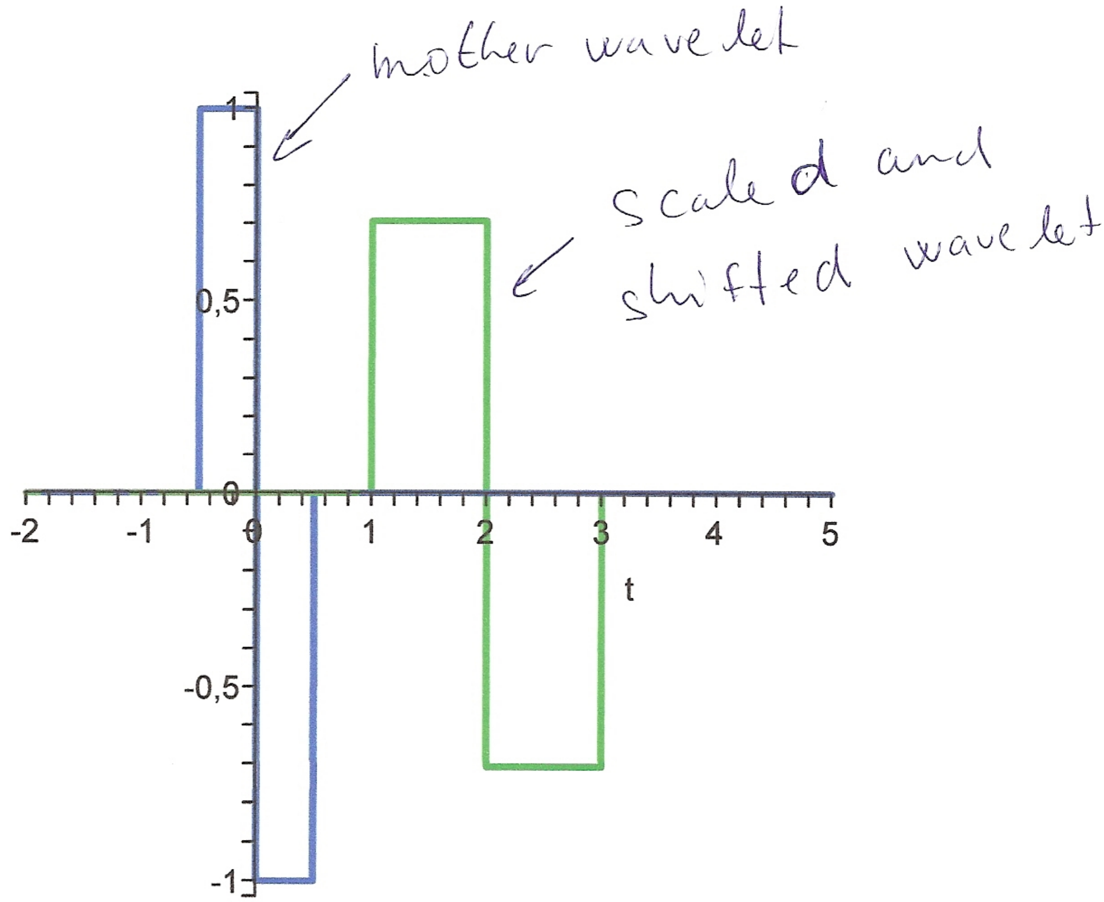

```
> wavelet(t):=piecewise((t-b)*w/2/Pi<-0.5,0,
                        (t-b)*w/2/Pi>=-0.5 and (t-b)*w/2/Pi<0,
  sqrt(w/2/Pi),
                        (t-b)*w/2/Pi>0 and (t-b)*w/2/Pi<=0.5,
  -sqrt(w/2/Pi),
```

(t-b)*w/2/Pi>0.5, 0);

220

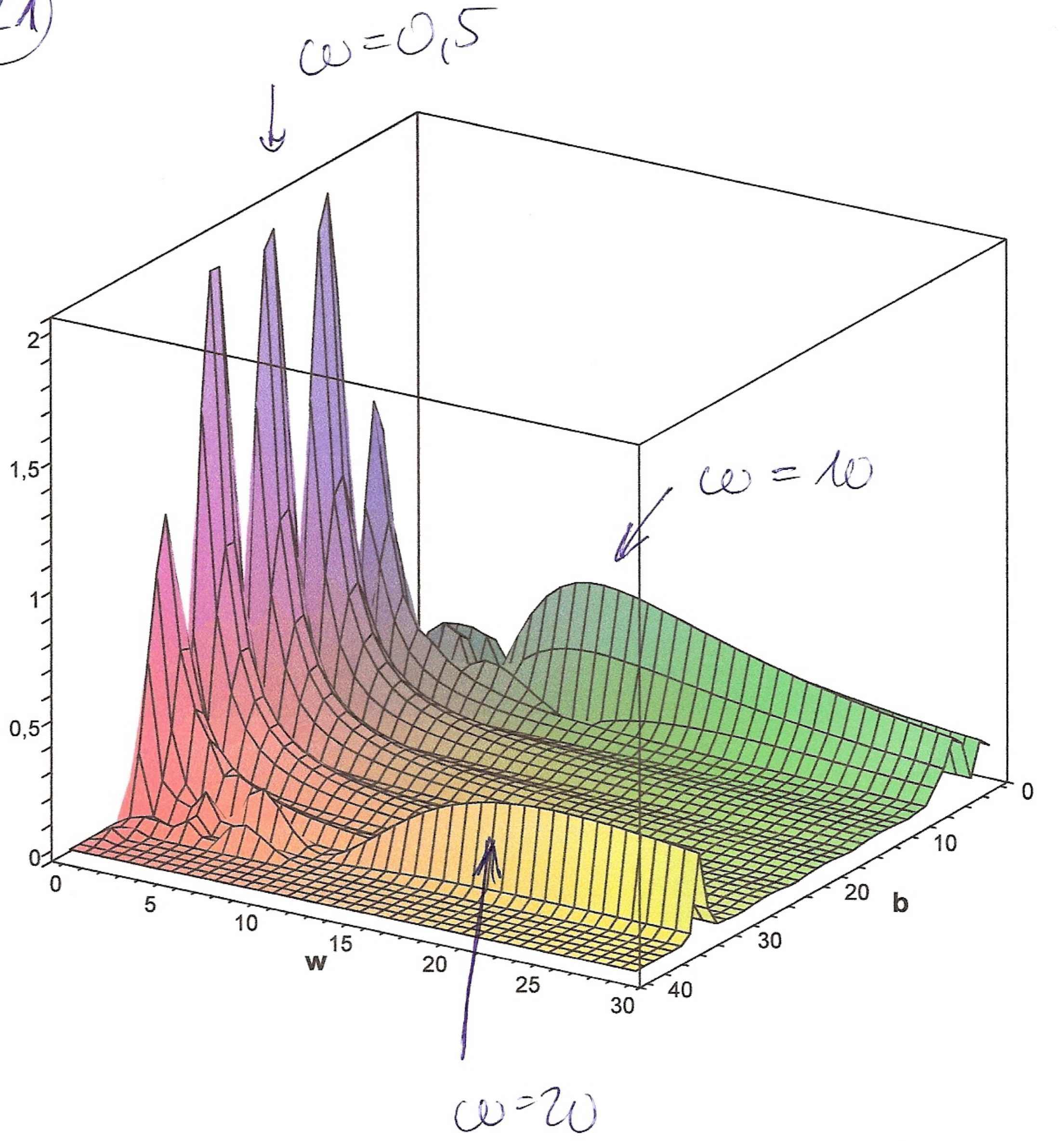
$$wavelet(t) := \begin{cases} 0 & \frac{(t-b)\,w}{2\,\pi} < -0.5 \\ \frac{1}{2}\sqrt{2}\,\sqrt{\frac{w}{\pi}} & -0.5 \le \frac{(t-b)\,w}{2\,\pi} \textbf{ and } \frac{(t-b)\,w}{2\,\pi} < 0 \\ -\frac{1}{2}\sqrt{2}\,\sqrt{\frac{w}{\pi}} & 0 < \frac{(t-b)\,w}{2\,\pi} \textbf{ and } \frac{(t-b)\,w}{2\,\pi} \le 0.5 \\ 0 & 0.5 < \frac{(t-b)\,w}{2\,\pi} \end{cases}$$

```
> p1:=plot(subs(b=0,w=2*Pi,wavelet(t)),t=-2..5,color=blue):
> p2:=plot(subs(b=2,w=0.5*2*Pi,wavelet(t)),t=-2..5,color=green):
> display(p1,p2);
```




```
> FW(w,b):=int(f(t)*wavelet(t),t=-2*Pi..13*Pi):
> FW2(w,b):=sqrt(Re(FW(w,b))^2+Im(FW(w,b))^2):
> plot3d(FW2(w,b),b= 0..40, w=0..30,
         style = patch,
         grid = [40,40],
         orientation = [5,25],
         axes=boxed,
         labelfont = [TIMES,BOLD,12]);
```

221



>

Hilbert transform

Let's consider the function $v(t)$.
Hilbert transform of $v(t)$ is defined as

$$\mathcal{H}\{v(t)\} = -\frac{1}{\pi t} * v(t) =$$

$$= -\frac{1}{\pi}\int_{-\infty}^{\infty} \frac{v(s)}{t-s}\, ds = \frac{1}{\pi}\int_{-\infty}^{\infty} \frac{v(s)}{s-t}\, ds =$$

$$= \frac{1}{\pi} \lim_{\varepsilon \to 0} \left(\int_{-\infty}^{t-\varepsilon} \frac{v(s)}{s-t}\, ds + \int_{t+\varepsilon}^{\infty} \frac{v(s)}{s-t}\, ds \right)$$

What's the meaning of $-\frac{1}{\pi t} * v(t)$?

Let's try Fourier transform of it:

$$\mathcal{F}\left\{ -\frac{1}{\pi t} * v(t) \right\} = \mathcal{F}\left\{ -\frac{1}{\pi t} \right\} \cdot \mathcal{F}\{v(t)\} =$$

$$= i \operatorname{sgn} \omega \cdot V(\omega)$$

$\Rightarrow$ Hilbert transform performs

"$\frac{\pi}{2}$" phase shift for $\operatorname{Re}\omega > 0$ and

"$-\frac{\pi}{2}$" phase shift for $\operatorname{Re}\omega < 0$.

It also means, that $\mathcal{H}\{\mathcal{H}\{v(t)\}\} = -v(t)$

i.e. $\mathcal{H}\{v(t)\} = \mathcal{H}^{-1}\{-v(t)\} =$

$= -\mathcal{H}^{-1}\{v(t)\}$.

<u>Remarks</u> Hilbert transform has important application in computing the envelope and mean of signal $v(t)$, $v(t) \in \mathbb{R}$. We define the so called analytic signal

$$\underline{z(t) = v(t) - i\,\mathcal{H}\{v(t)\}}.$$

The amplitude $|z(t)|$ represents the envelope of signal $v(t)$, see the examples:

```
> with(inttrans):with(plots):
```

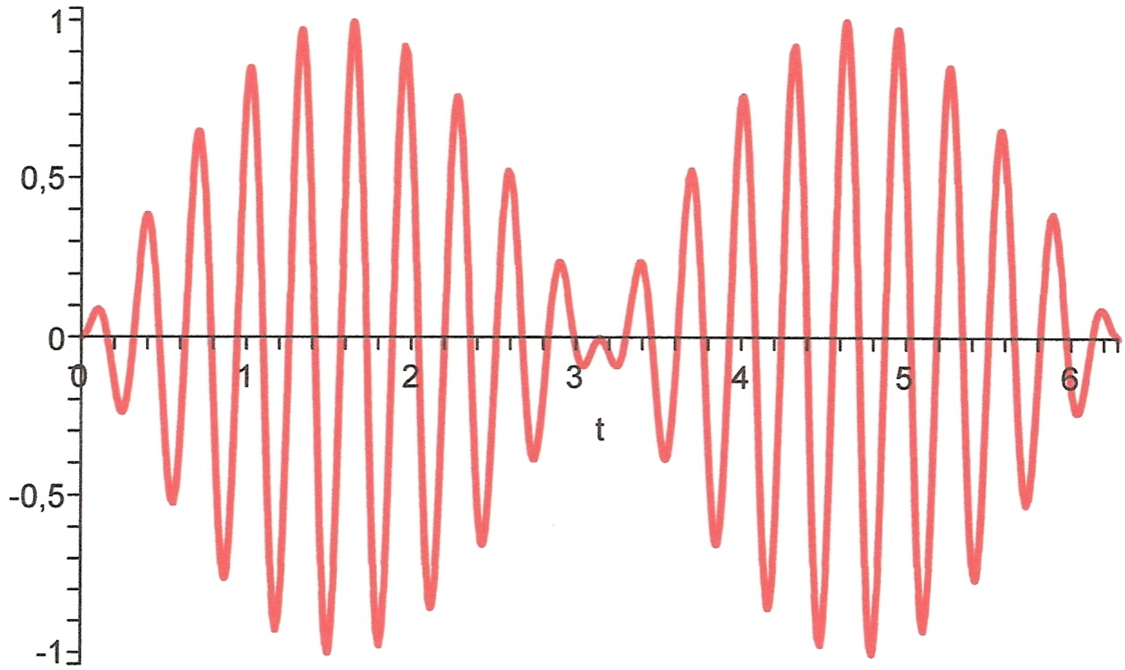
Warning, the name changecoords has been redefined

```
> f(t):=sin(t)*sin(20*t);
```

$$f(t) := \sin(t)\sin(20\,t)$$

original signal $f(t)$

```
> plot(f(t),t=0..2*Pi);
```




```
> g(s):=hilbert(f(t),t,s);
```

$$g(s) := -\frac{1}{2}\sin(19\,s) + \frac{1}{2}\sin(21\,s)$$

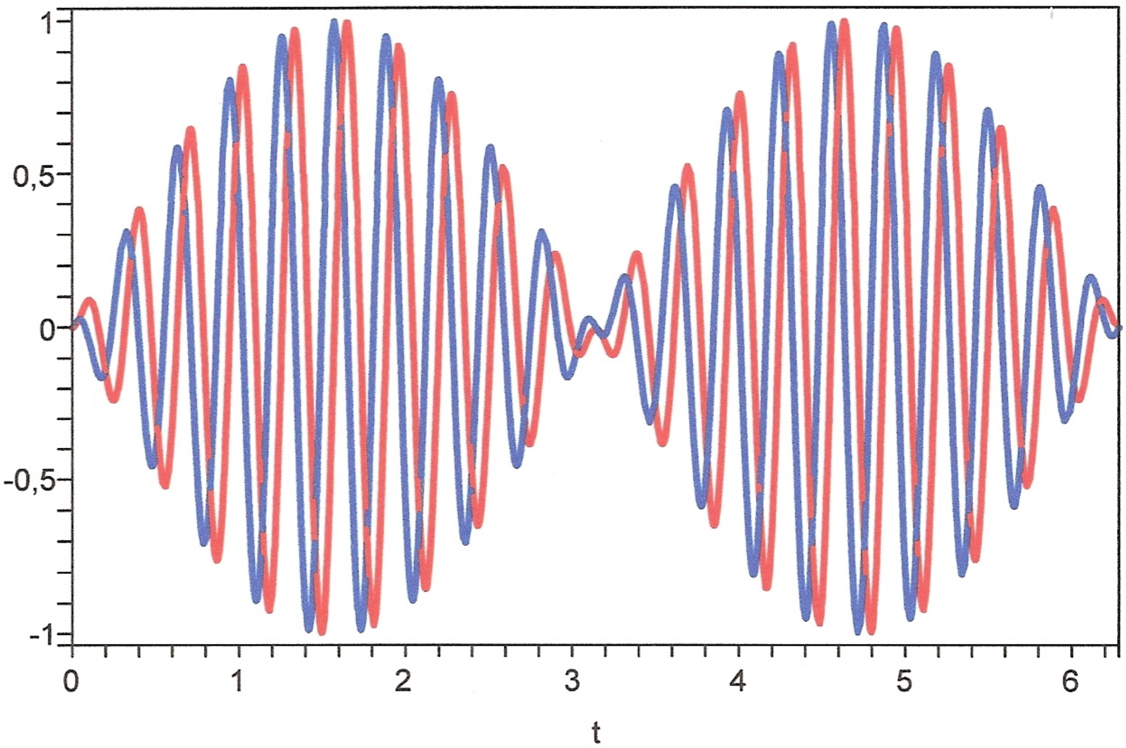
```
> g(t):=subs(s=t,g(s));
```

$$g(t) := -\frac{1}{2}\sin(19\,t) + \frac{1}{2}\sin(21\,t)$$

$\mathcal{H}\{f(t)\}$

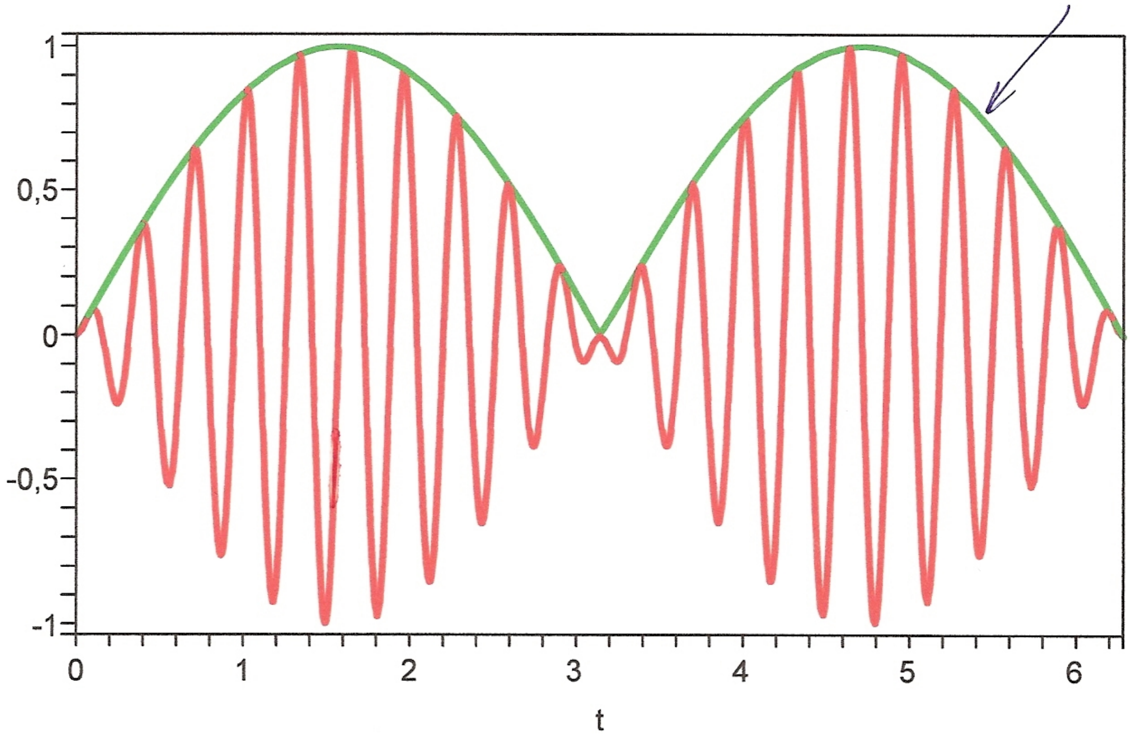
```
> p1:=plot(f(t),t=0..2*Pi,color=red):
> p2:=plot(g(t),t=0..2*Pi,color=blue):
> display({p1,p2},axes=boxed);
```

225




```
> p3:=plot(sqrt(f(t)^2+g(t)^2),t=0..2*Pi,color=green):
> display({p1,p3},axes=boxed);
```

$|z(t)|$




```
>
```

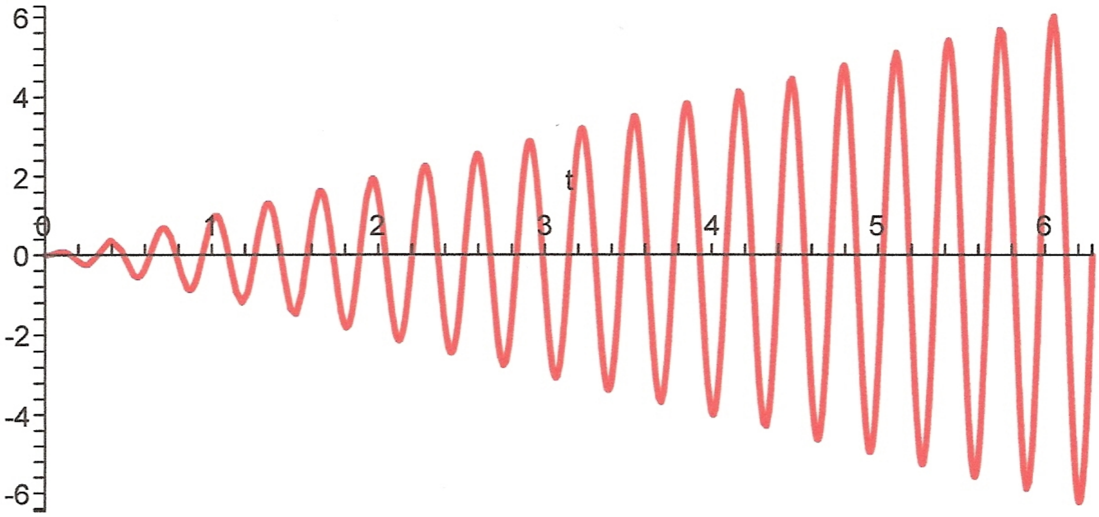
Another example

```
> with(inttrans):with(plots):
> f(t):=t*sin(20*t);
```

$$f(t) := t\sin(20\,t)$$

```
> plot(f(t),t=0..2*Pi);
```

f(t)




```
> g(s):=hilbert(f(t),t,s);
```

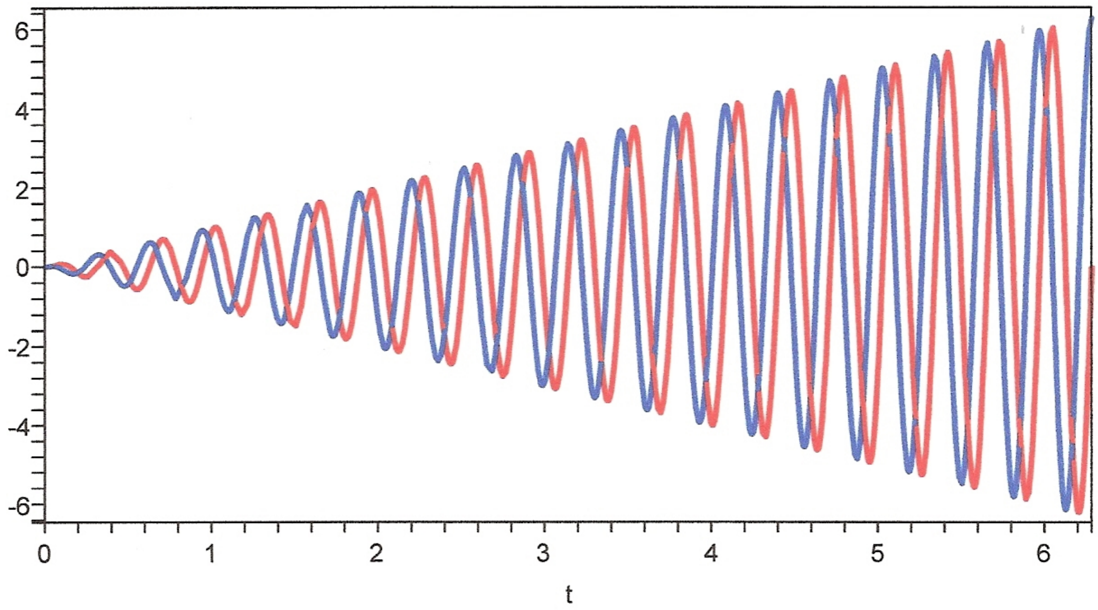
$$g(s) := s\,(524288\cos(s)^{20} - 2621440\cos(s)^{18} + 5570560\cos(s)^{16} - 6553600\cos(s)^{14} + 4659200\cos(s)^{12} - 2050048\cos(s)^{10} + 549120\cos(s)^{8} - 84480\cos(s)^{6} + 6600\cos(s)^{4} - 200\cos(s)^{2} + 1)$$

```
> g(t):=subs(s=t,g(s));
```

$$g(t) := t\,(524288\cos(t)^{20} - 2621440\cos(t)^{18} + 5570560\cos(t)^{16} - 6553600\cos(t)^{14} + 4659200\cos(t)^{12} - 2050048\cos(t)^{10} + 549120\cos(t)^{8} - 84480\cos(t)^{6} + 6600\cos(t)^{4} - 200\cos(t)^{2} + 1)$$

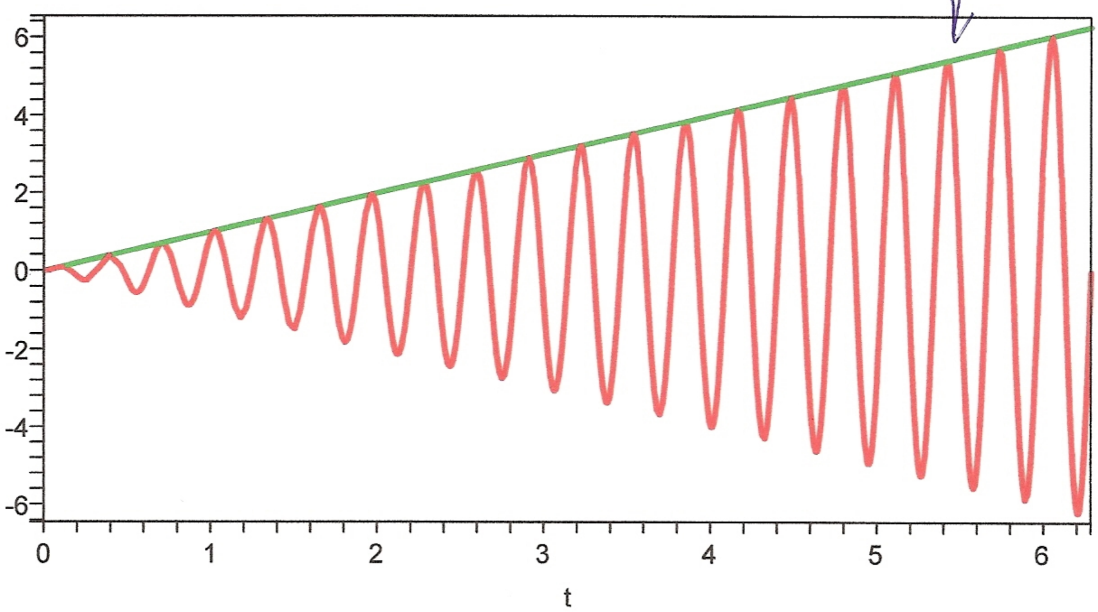
$\mathcal{H}\{f(t)\}$

```
> p1:=plot(f(t),t=0..2*Pi,color=red):
> p2:=plot(g(t),t=0..2*Pi,color=blue):
> display({p1,p2},axes=boxed);
```




```
> p3:=plot(sqrt(f(t)^2+g(t)^2),t=0..2*Pi,color=green):
> display({p1,p3},axes=boxed);
```

$|z(t)|$




```
>
>
```

(228)

Hilbert - Huang transform

Step 1. Envelope the original data, X(t). The envelope is determined by the maximum and minimum of X(t).

Step 2. Determine the local mean, m_0.

Step 3. Subtract the mean from the data to determine the first IMF, h_{1n}. Then determine the first residual value, r_1.

$$h_{1k} = X(t) - m_0$$

$$r_1 = X(t) - h_{1n}$$

Step 4. Repeat until final IMF, h_{nk}, is found. The summation of the IMF and residual should return the original data.

$$X(t) = \sum h_{1\text{-}nk} + r_n$$

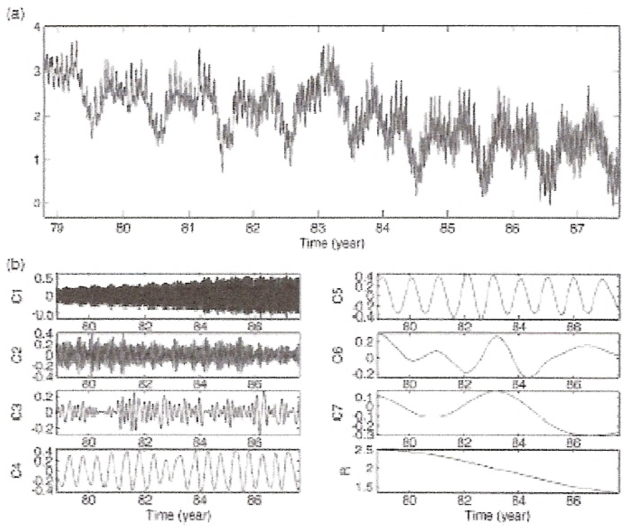


Figure 1. MF Decomposition (Vatchev, 2002)

Summary of signal analysis

	Fourier transform	Wavelet transform	Hilbert-Huang transform
base functions	apriori	apriori	adaptive (IMF) - result of "computation"
frequency spectr	global	local - some errors may appear	local - no errors
nelinear signal	no	no	yes
nonstationary signal	no	yes	yes
theory	yes	yes	only empirical