

Beispiel zur Schnellen Fourier-Transformation

```
> x := Array([-1,-1,-1,-1,1,1,1,1]); # real parts of data  
x := [-1, -1, -1, -1, 1, 1, 1, 1]
```

```
> y := Array([0,0,0,0,0,0,0,0]); # imaginary parts of data  
y := [0, 0, 0, 0, 0, 0, 0, 0]
```

```
> FFT(3,x,y): # transform data
```

```
> x;  
[0, -2.000000001, 0., -1.999999999, 0, -1.999999999, 0., -2.000000001]
```

```
> y; # imaginary parts of transformed data  
[0, 4.828427122, 0., 0.828427124, 0, -0.828427124, 0., -4.828427122]
```

```
> zip((a,b)->a+b*I, x, y): convert(%, list);  
[0, -2.000000001 + 4.828427122 I, 0. + 0. I, -1.999999999 + 0.828427124 I, 0, -1.999999999 - 0.828427124 I,  
0. + 0. I,  
-2.000000001 - 4.828427122 I]
```

```
> iFFT(3,x,y): # check results
```

```
> x;  
[-1.000000000, -0.999999999, -0.999999995, -0.999999985, 1.000000000, 0.999999990, 0.999999995,  
0.999999985]
```

```
> y;  
[0., 2.500000000 10-10, 0., -2.500000000 10-10, 0., -2.500000000 10-10, 0., 2.500000000 10-10]
```

```
> y := map(fnormal, y);  
y := [0., 0., 0., -0., 0., -0., 0., 0.]
```

Glättung einer verrauschten harmonischen Schwingung durch Faltung mit einer Gaussverteilung

Definition einer Störfunktion (Rauschen)

```
> noise := stats[random, normald]:
```

Harmonische Schwingung mit überlagertem Rauschen

```
> re_data := Array([seq(sin(0.0625*k)+0.1*noise(),  
k=1..2^8)]);
```

```
re_data := [ 256 Array  
Data Type: anything  
Storage: rectangular  
Order: Fortran_order ]
```

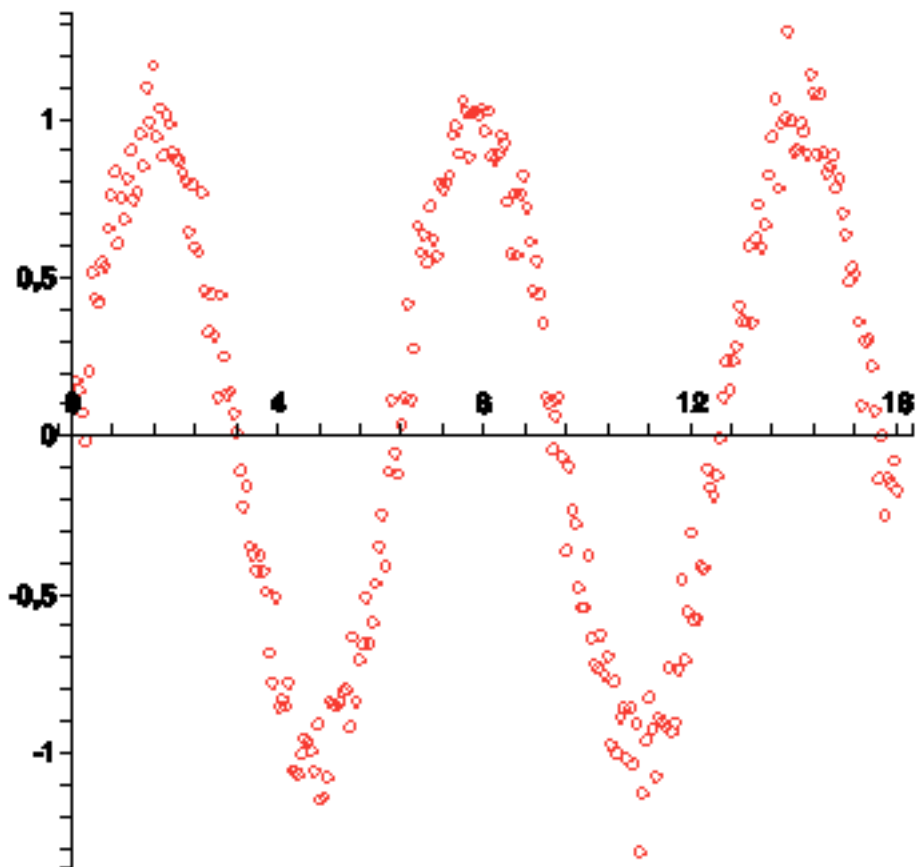
```
> im_data := Array([seq(0, k=1..2^8)]):
```

```
> xcoords := Array([seq(0.0625*k, k=1..2^8)]):
```

Grafische Darstellung

```
> plotdata := convert(zip((a,b)->[a,b], xcoords,  
re_data), list):
```

```
> plot(plotdata, style=POINT);
```



[Definition einer Gaussfunktion zur Faltung (Kern,kernel)

```
> re_kernel := Array([seq(exp(-100.0*(k/2^8)^2),  
  k=1..2^8)]):
```

```
> im_kernel := Array([seq(0, k=1..2^8)]):
```

[FFT in den Frequenzbereich von verrauschten Daten und Kern

```
> FFT(8, re_data, im_data):
```

```
> FFT(8, re_kernel, im_kernel):
```

[Punktweise Multiplikation der transformierten Daten

```
> data := zip((a,b)->(a+b*I), re_data, im_data):
```

```
> kernel := zip((a,b)->(a+b*I), re_kernel, im_kernel):
```

```
> newdata := zip((a,b)->a*b, data, kernel):
```

```
> new_re_data := map(Re, newdata):
```

```
> new_im_data := map(Im, newdata):
```

[Rücktransformation mittels inverser FFT

```
> iFFT(8, new_re_data, new_im_data):
```

[Grafische Darstellung der geglätteten Daten

```
> plotdata := convert(zip((a,b)->[a,b], xcoords,  
  new_re_data), list):
```

```
> plot(plotdata, style=POINT);
```

