# Oefening: Discrete fourier transform

```
> restart: with(plots):with(DiscreteTransforms):
```

## Analytische uitdrukking voor

```
> part    := (k, N, ll) -> limit((exp(Pi*I*(l-k))/exp(Pi/N*I*(l-k)
  )) * (sin(Pi*(l-k))/ sin(Pi/N*(l-k))) + (exp(-Pi*I*(l+k))/exp(-
  Pi/N*I*(l+k))) * (sin(Pi*(l+k))/ sin(Pi/N*(l+k))),l=ll ):
> X_an := (k, N, ll) -> evalf(1/(2*sqrt(N))*part(k, N, ll)):
```

## Periodieke functie

We zullen discrete samples nemen van de cosinus functie met een bepaalde frequentie f

```
> T := 1:
> g :=(t,f) -> cos(2*Pi*f*t);
```

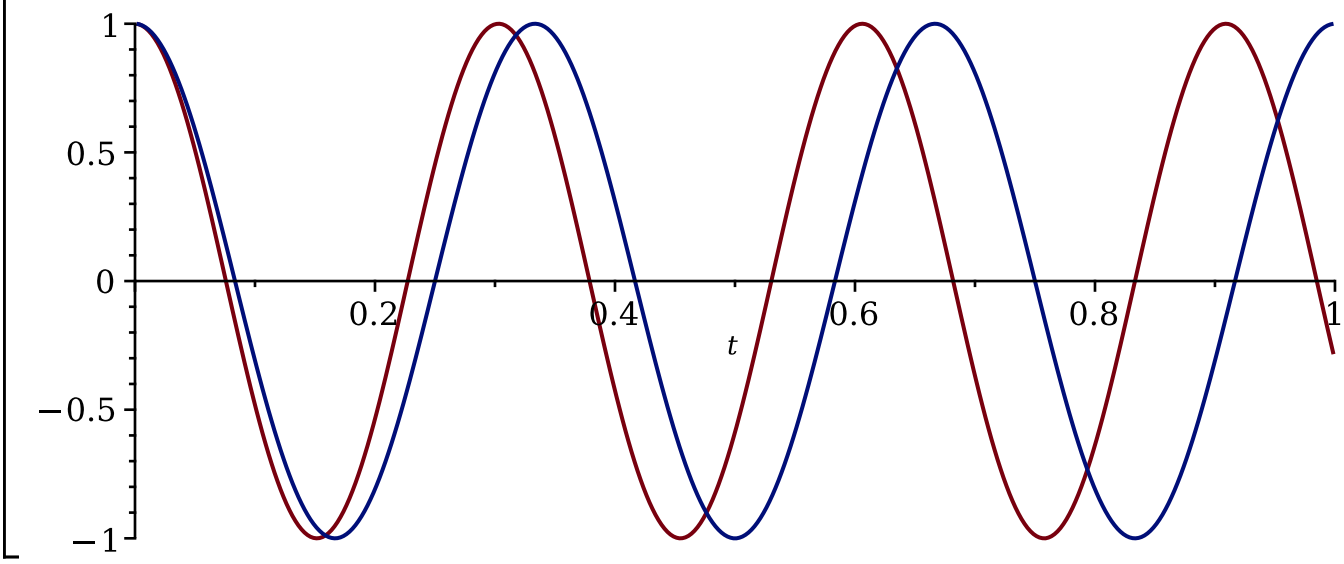
$$g := (t, f) \mapsto \cos(2 \cdot \pi \cdot f \cdot t) \quad (1)$$

We nemen 2 frequenties f1 en f2

```
> f1 := 3/T;
  f2 := 3.3/T;
```

$$f1 := 3$$
$$f2 := 3.3 \quad (2)$$

```
> plot({g(t,f1), g(t,f2)},t=0..T);
```




## Sampling: N samples

We definieren het aantal samples, de sampling frequentie en sampling periode:

```
> N  := 20;
  fS := N/T;
  dt := 1/fS;
```

$$N := 20$$
$$fS := 20$$

$$dt := \frac{1}{20} \quad (3)$$

We steken de discrete sample punten in een lijst

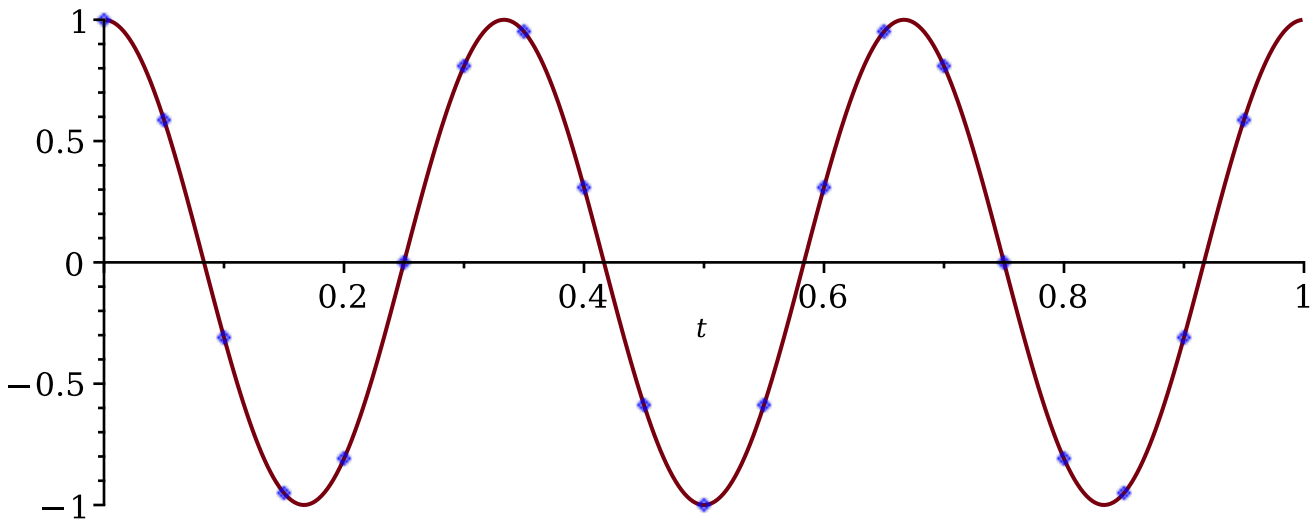
```
> samples1 := [seq(g(i*dt, f1), i=0..N-1)]:
```

En bepalen de **discrete fourier transform** van de discrete datapunten

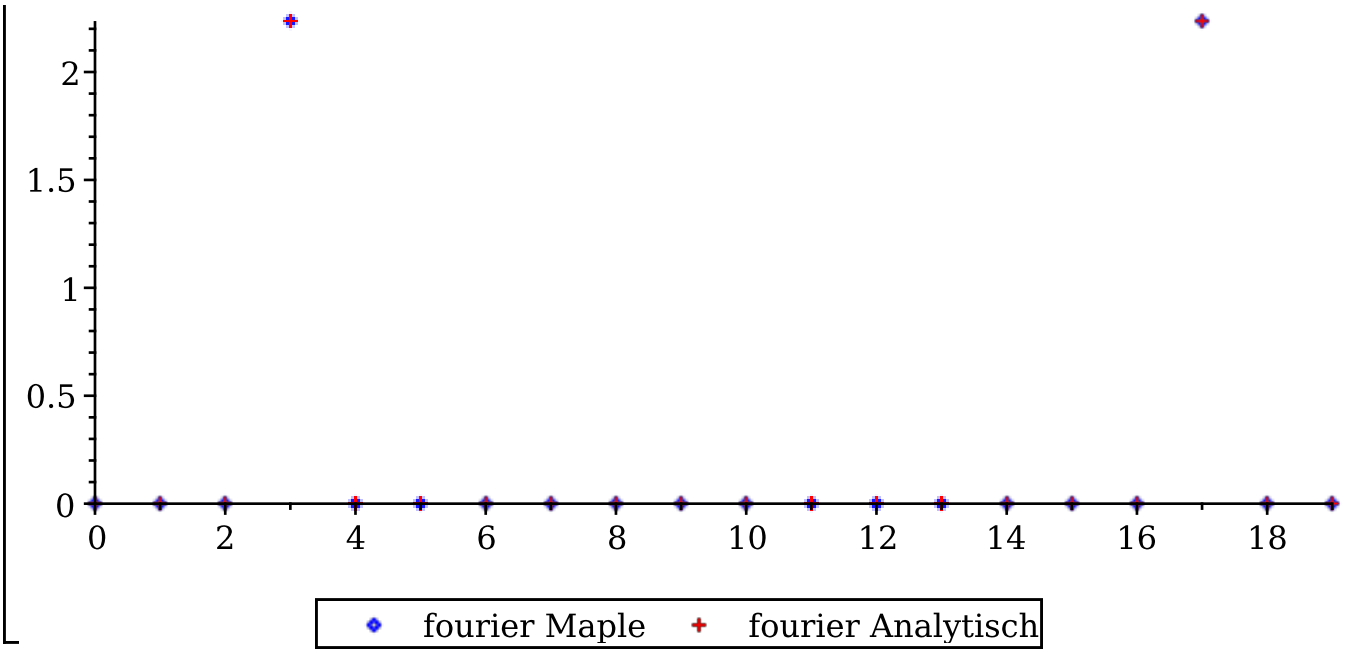
```
> F1      := FourierTransform(Vector(samples1)):
  F1_an   := [seq(X_an(k, N, 3), k=0..N-1)]:
```

Visualisatie:

```
> samplesplot1 := pointplot({seq([i*dt-dt,samples1[i]],i=1..N)},
  color=blue):
  plot1        := plot(g(t,f1),t=0..1):
  display(plot1, samplesplot1 );
```

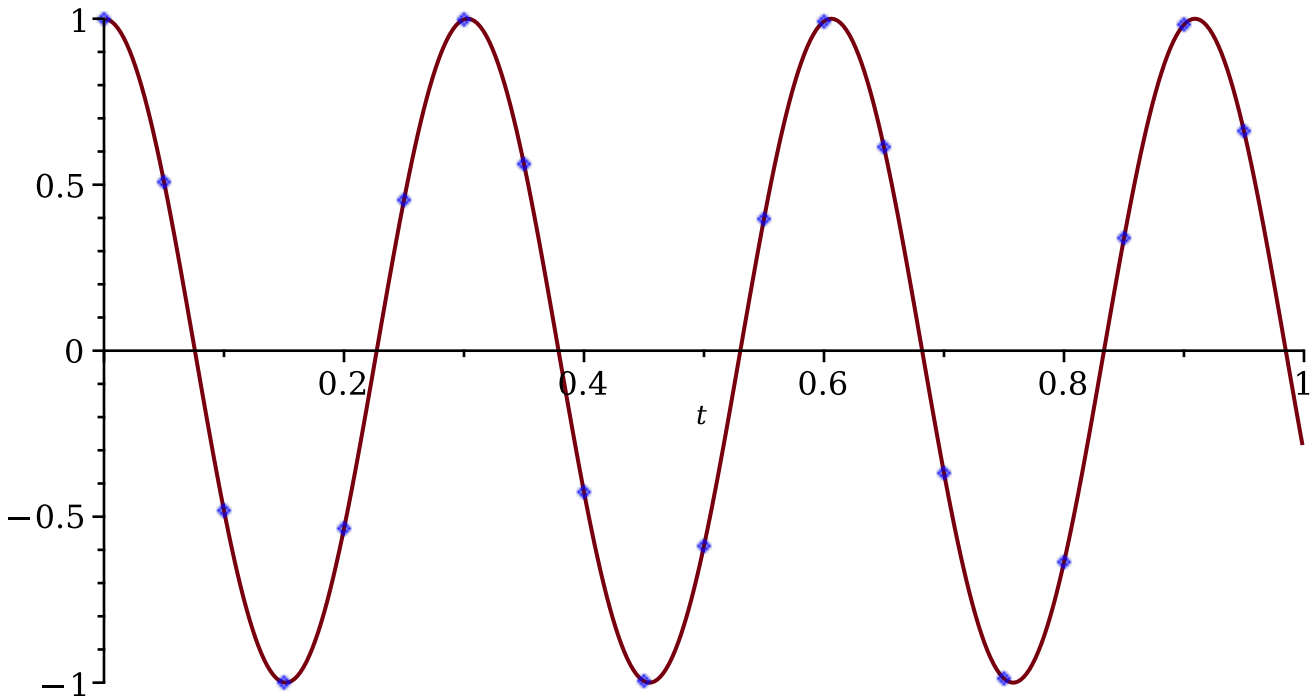



```
> fourierplot1     := pointplot({seq([i-1,abs(F1[i])],i=1..N)},
  color=blue   ,legend="fourier Maple"):
  fourier_an_plot1 := pointplot({seq([i-1,abs(F1_an[i])],i=1..N)},
  color=red, legend="fourier Analytisch",symbol=cross):
  display(fourierplot1, fourier_an_plot1);
```



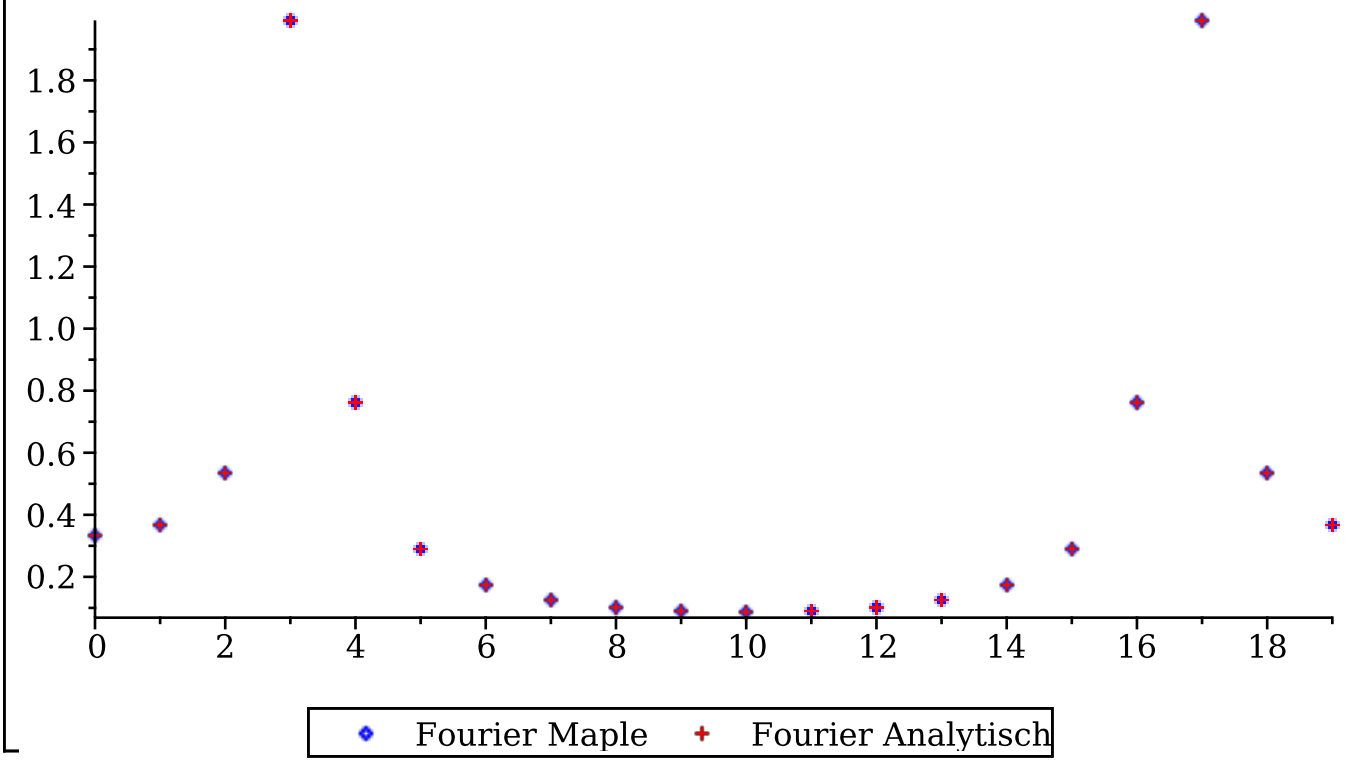

Doe hetzelfde voor een sample de frequentie f2

```
> samples2    := [seq(g(i*dt, f2), i=0..N-1)]:
> F2          := FourierTransform(Vector(samples2)):
> F2analytic  := [seq(X_an(k, N, 3.3), k=0..N-1)]:
> samplesplot2 := pointplot({seq([i*dt-dt,samples2[i]],i=1..N)},
  color=blue):
  plot2 := plot(g(t,f2),t=0..1):
  display(plot2, samplesplot2);
```




```
> fourierplot2    := pointplot({seq([i-1,abs(F2[i])],i=1..N)},
  color=blue      , legend="Fourier Maple"):
  fourier_an_plot2 := pointplot({seq([i-1,abs(F2analytic[i])],i=1..
```

```
N)},color=red, legend="Fourier Analytisch",symbol=cross):
display(fourierplot2, fourier_an_plot2);
```




# Aliasing

Neem nu de frequentie f3=12.1/T, groten dan de nyquist frequentie: f_N = f_S/2

```
> fN:=fS/2;
```

$$fN := 10 \quad (4)$$

```
> f3 := 12.1/T;
```

$$f3 := 12.1 \quad (5)$$

```
> samples3     := [seq(g(i*dt,f3), i=0..N-1)]:
  samples_alias := [seq(g(i*dt,fS-f3), i=0..N-1)]:
```

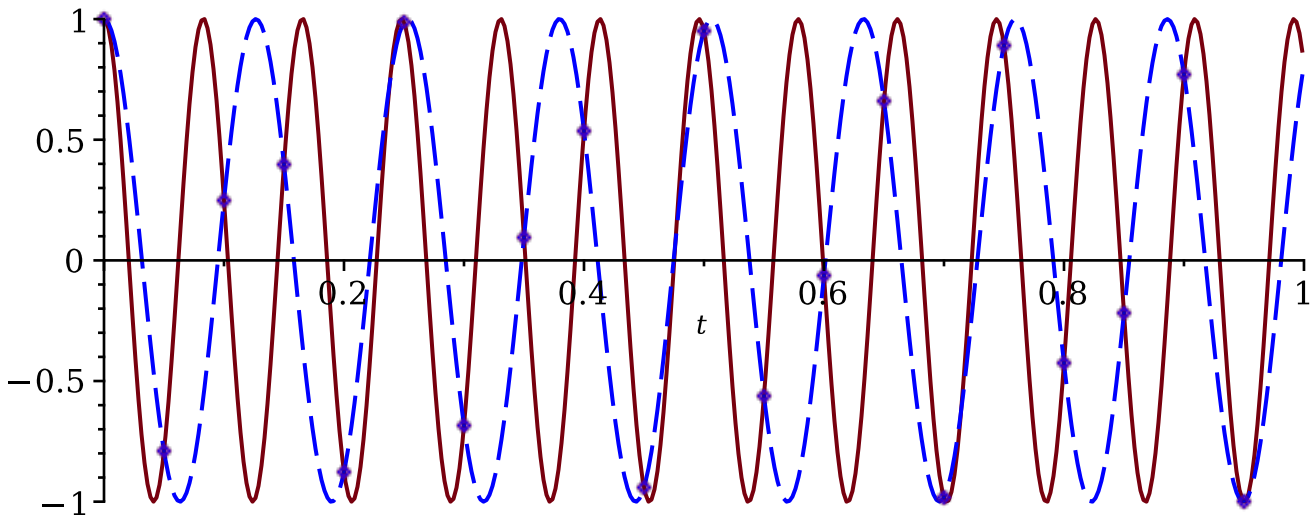
En bepalen de **discrete fourier transform** van de discrete datapunten

```
> F3      := FourierTransform(Vector(samples3)):
> Falias  := FourierTransform(Vector(samples_alias)):
```

Visualisatie:

```
>
  samples_alias_plot  := pointplot({seq([i*dt-dt,samples3[i]],i=1..
  N)},color=blue):
  samples_plot        := pointplot({seq([i*dt-dt,samples_alias[i]
  ],i=1..N)},color=red):
  plot3               := plot(g(t,f3),t=0..1):
  plotalias           := plot(g(t,fS-f3),t=0..1, color=blue,
  linestyle="dash"):
```

```
display(plot3 ,plotalias, samples_plot, samples_alias_plot);
```




```
> spec1    := pointplot({seq([i-1,abs(F3[i])],i=1..N)},    color=
  blue, legend="f=14.2 spectrum"):
  aliasspec := pointplot({seq([i-1,abs(Falias[i])],i=1..N)},color=
  red, legend="alias spectrum",symbol=cross):
  display(spec1, aliasspec);
```