

FFT and IFFT in Maple and Matlab

Nasser M. Abbasi

August 2005

Compiled on May 21, 2020 at 1:59am

This example show how to do FFT on a continouse time function, then do an IFFT to recover the original function. This is done in Matlab and Maple

1 Matlab

```
1 clear all;
2 t = linspace(-pi,pi,100);
3 dt = t(2)-t(1);
4 fs = 1/dt;
5 w = linspace(0,fs,100);
6
7 y = sin(t) + 2*sin(3*t);
8 g = exp(-w) .* fft(y);
9 h = ifft(g);
10 plot(real(h));
```

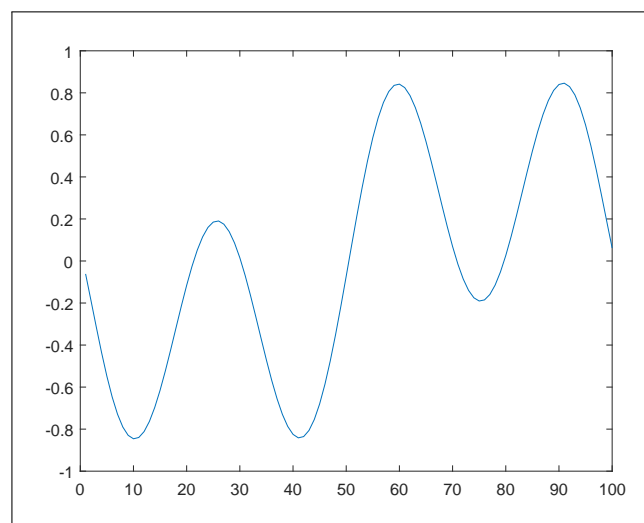


Figure 1: Output of the above

2 Maple

```

1 #showing how to do FFT and IFT in maple 10. Nasser M. Abbasi.
2 restart;
3 with(inttrans):
4 with(LinearAlgebra):
5 with(plots):
6 N:=100:
7
8 #the function to fft
9 y:= t->sin(t) + 2*sin(3*t):
10
11 linspace:=proc(fromP,toP,n)
12     local incr,data,i,T,L;
13     incr:=(toP-fromP)/(n-1);
14     L:=Vector(1..n,[]);
15     T:=fromP;
16     for i from 1 to n do
17         L[i]:=T;
18         T:=T+incr;
19     end do;
20     return(L);
21 end proc:
22
23 data:=Vector(1..N,[]):
24 g:=Vector(1..N,[]):
25 T:=linspace(-Pi,Pi,N):
26
27 for i from 1 to N do
28     data[i]:=evalf(y(T[i]));
29 end do:
30
31 Y:=DiscreteTransforms:-FourierTransform( data,algorithm=DFT,padding=0 ):
32 adj:=evalf(sqrt(1/N)): #adjust as maple has normalization factor
33 Y:=Y/adj:
34
35 dt:=T[2]-T[1]:
36 fs:=1/dt:
37 w:=evalf(linspace(0,fs,N)):
38 s:=[seq(exp(-w[i]),i=1..N)]:
39 g:=Vector(N,zip((x,y)->x*y,s,convert(Y,list))):
40 h:=DiscreteTransforms:-InverseFourierTransform( g ):
41 h:=h*sqrt(1/N):#adjust the IFFT due to maple normalization again
42
43 listplot(map(Re,h),color=red,labels=["t","y(t)"],title="result of IFFT");

```

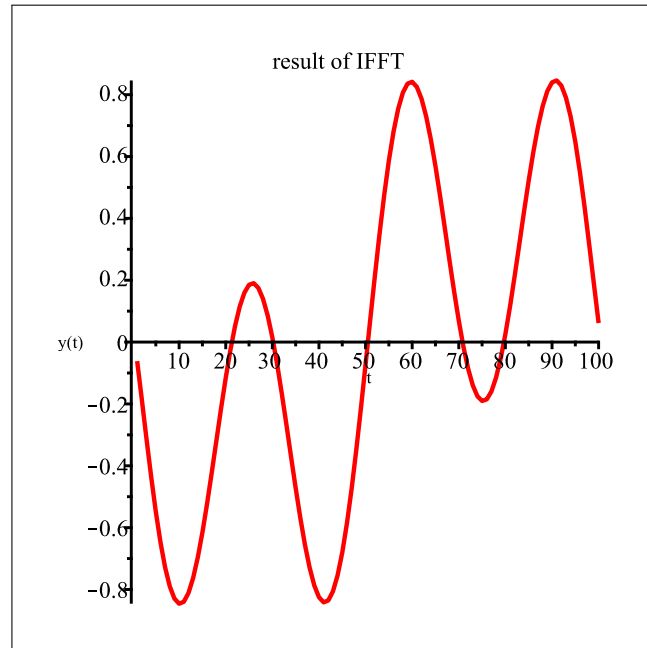


Figure 2: Output of the above

3 Source code download

1. Matlab `fft_example.m`
2. Maple `maple.txt`