
Subject: Re: Using IDL to make a signal filter
Posted by [Wox](#) on Tue, 15 Jul 2008 09:32:48 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Mon, 14 Jul 2008 23:21:33 -0700 (PDT), ICBM0926
<ICBM0926@gmail.com> wrote:

```
<snip>
> kxcen=7.757e+6
> kxband=7.757e+6
> NyKx=2.d*3.1415926/2.d*dx
> mkx=dindgen(2048)*dkx
> mask=dindgen(2048)
> mask=1.d*(exp(-((mkx-kxcen)/(kxband/2.d))^8.d)+exp(-((mkx-(2.d*NyKx-
> kxcen))/(kxband/2.d))^8.d))
<snip>
```

The filter above removes all negative frequencies. This is why you end up with half the amplitude. Check fft documentation: negative frequencies in the second half, so the second half of the filter (in 1/m space) should be a (reversed) copy of the first half. See below how it's done. I got confused by your notation, so I transformed things a bit.

```
pro onedanalytic
; constants
lambda=810.0e-9
c=299792458.d
n=2049
dx=800.0e-9/32.d
E0=1.d

; (m)-domain
k=2*dpi/lambda
x=dx*(dindgen(n)-n/2)
E1=E0*exp(-(x/(3.822e-14*c))^2)*cos(k*x)
E2=E0*exp(-(x/(3.822e-14*c))^2)*cos(2.d*k*x)
E=E1+E2

; (1/m)-domain
Efft=fft(E,/double)
nkx=n/2+1
kx=dindgen(nkx)/(n*dx)
kx=[kx,reverse(-kx[1:nkx-1-(~(n mod 2))])]

; filter
mask=exp(-(2*(kx*lambda-1))^8.d)+exp(-(2*((kx-dx)*lambda+1))^8.d)
```

```
Efft*=mask
ReE=fft(Efft,/inverse)

; plot
window,0
!P.Multi=[0,2,2]
plot,x,E1,title='E1'
plot,kx,mask,title='Filter: remove E2'
plot,x,ReE,title='Filtered E1+E2 = E1'
end;pro onedanalytic
```
