Subject: Re: Curve Fitting to timeseries using a set of 8 sine and cosine functions
Posted by Yngvar Larsen on Sat, 01 Nov 2014 14:57:24 GMT
View Forum Message <> Reply to Message

On Saturday, 1 November 2014 04:12:03 UTC+1, siumt...@gmail.com wrote:

> My question is not resolved yet.
>
> Craig: Why you did not suggest me to use your MPFIT code for fitting my model.

Craig can answer for himself, but the answer is likely to be that a nonlinear optimisation is not neccesary to solve a linear problem. The least squares problem can be solved with linear algebra in your case (multiple linear regression).

> Yngvar: I think you have helped me enough and spend a lot of time on my problem. You seem to fed up of my question because I did not understand you.

Well. I don't know what more to help you with. Your signal does not fit your model, as can clearly be seen from a simple harmonic analysis, which you can adapt to the irregular case by either

1) subtracting the mean and replacing the missing samples with zero, as Craig suggested, or

2) if you have a lot of missing data or non-equispaced sampling, a nonuniform FFT, often referred to as NFFT or NUFFT, e.g.

http://doi.acm.org/10.1145/1555386.1555388

Running the code I posted earlier will show you that you indeed have a peak of some power exactly at the frequency with a period of 12 months.

****
EDIT: looking again at the model in your original post instead of your code, I see that you have a factor 2 difference in your code for the model frequencies: 2*!pi*n/12 in the code vs pi*n/12 in the model in your original post. This means your variable ANGLE should be changed to ANGLE = 2*!dpi*t[g]/12, and you will get different results (better fit).

Assuming the model in the original post is correct, I get a better fit to your signal using frequency components (2*12)/n months, n=1,2,3,4 (24, 12, 8, and 6 months periods).

```
8<--------------------------------------------
datafile = 'testdata.txt'
np = file_lines(datafile)
data = strarr(np)
openr, unit, datafile, /get_lun
readf, unit, data
free_lun, unit
data = double(data[*])

;; Subtract mean
```

```
data -= mean(data)

;; Time and freq axes
dt = 1/12d0                     ; [years]
t = dindgen(np)*dt
df = 1/(np*dt)
faxis = (1+dindgen(np/2))*df

nc = 4
f = (1+dindgen(nc))/24

;; Generate signal according to model
H = dblarr(np, 2*nc)            ; System matrix
for n=0,nc-1 do begin
   H[*,n] = cos(2*!dpi*f[n]*t)     ; even terms
   H[*,n+4] = sin(2*!dpi*f[n]*t)   ; odd terms
endfor

;; least squares fit to signal:
Hpinv = invert(transpose(H)#H)#transpose(H) ; pseudoinverse of linear system
coeff_est = Hpinv#data
;; Fitted signal
s_fit = H#coeff_est

plot, t, data
oplot, t, s_fit, col=255

8<---------------------------------------------
```

This can all be done more easily with the REGRESS function, with goodness-of-pit parameters included in the deal. But a signal processing freak, I like to control the details :)

Over and out,

--
Yngvar