Subject: Curve Fitting to timeseries using a set of 8 sine and cosine functions Posted by siumtesfai on Sat, 25 Oct 2014 05:47:15 GMT

View Forum Message <> Reply to Message

I think you should try to be for specific to ask question here.

Suppose I have a timeseries with the S size.

I want to do nonlinear fitting to the timeseries using the following fourier series ( harmonic function)

And I would find 8 coefficients such as An and Bn where n = 1,2,3,4

That is.

A1,A2,A3,A4 B1,B2,B3,B4

I have attempted to understand how it works mpfit by Craig and curvefit. Unfortunately, I did not because I am not IDL expert. So I posted this if anyone can help

**Best Wishes** 

Thanks for you help

Subject: Re: Curve Fitting to timeseries using a set of 8 sine and cosine functions Posted by Yngvar Larsen on Sat, 25 Oct 2014 15:59:25 GMT View Forum Message <> Reply to Message

This is actually not a nonlinear system, but a linear one. Thus, in the general case where your sampling vector X is not regular, a linear least squares fit could be done easily with the pseudo inverse of the system matrix:

```
np = 160 ; Number of samples
nc = 4 ; Number of even/odd terms

;; Irregular sampling points
x = 2*!dpi*randomu(seed, np)-!dpi
x = x[sort(x)]

;; Generate signal according to model
H = dblarr(np, 2*nc+1) ; System matrix
H[*,0] = 1 ; Constant term
```

```
for n=1,nc do begin
 H[*,n] = cos(n*x)
                          ; even terms
 H[*,n+nc] = sin(n*x)
                         ; odd terms
endfor
coeff = randomn(seed, 2*nc+1); Random coefficents
s = H\#coeff
                        ; signal
n = randomn(seed, np)
                             ; noise
:: least squares fit to signal:
Hpinv = invert(transpose(H)#H)#transpose(H); pseudoinverse of linear system
coeff est = Hpinv#s
print, 'RMS: ', sqrt(mean(abs(coeff_est - coeff)^2)); Exact within numerical precision
;; least squares estimate of system coefficients
coeff_est = Hpinv#(s+n)
print, 'RMS: ', sqrt(mean(abs(coeff_est - coeff)^2))
;; Fitted signal
s fit = H\#coeff est
plot, x, s+n, linestyle=1, thick=2.; Noisy observation
oplot, x, s, color='ff'x
                      : True signal
oplot, x, s_fit, color='ff00'x
                              : Fitted signal
8<-----
If your sample vector X happens to be regular, the solution to your problem is actually nothing
more than an FFT, and pick the 5 first complex coefficients. The first coefficient is the constant
term A0 (not included in your problem), and real/imaginary parts of the following coefficients
corresponds to cosine terms A1-A4 and the sine terms B1-B4, respectively.
Yngvar
On Saturday, 25 October 2014 07:47:17 UTC+2, siumt...@gmail.com wrote:
> I think you should try to be for specific to ask question here.
>
> Suppose I have a timeseries with the S size.
> I want to do nonlinear fitting to the timeseries using the following fourier series (harmonic
function)
>
 And I would find 8 coefficients such as An and Bn where n = 1,2,3,4
> That is.
```

```
> A1,A2,A3,A4
> B1,B2,B3,B4
> I have attempted to understand how it works mpfit by Craig and curvefit . Unfortunately, I did not because I am not IDL expert. So I posted this if anyone can help
> Best Wishes
> Thanks for you help
```

Subject: Re: Curve Fitting to timeseries using a set of 8 sine and cosine functions Posted by siumtesfai on Sat, 25 Oct 2014 17:33:06 GMT

View Forum Message <> Reply to Message

On Saturday, October 25, 2014 11:59:28 AM UTC-4, Yngvar Larsen wrote:

> This is actually not a nonlinear system, but a linear one. Thus, in the general case where your sampling vector X is not regular, a linear least squares fit could be done easily with the pseudo inverse of the system matrix:

```
> np = 160
                         ; Number of samples
> nc = 4
                        : Number of even/odd terms
> ;; Irregular sampling points
> x = 2*!dpi*randomu(seed, np)-!dpi
> x = x[sort(x)]
> ;; Generate signal according to model
> H = dblarr(np, 2*nc+1)
                              ; System matrix
> H[*,0] = 1
                         ; Constant term
> for n=1,nc do begin
   H[*,n] = cos(n*x)
                           ; even terms
    H[*,n+nc] = sin(n*x)
                            ; odd terms
> endfor
> coeff = randomn(seed, 2*nc+1); Random coefficents
> s = H\#coeff
                          ; signal
> n = randomn(seed, np)
                             ; noise
> ;; least squares fit to signal:
> Hpinv = invert(transpose(H)#H)#transpose(H); pseudoinverse of linear system
> coeff est = Hpinv#s
 print, 'RMS: ', sqrt(mean(abs(coeff_est - coeff)^2)); Exact within numerical precision
> ;; least squares estimate of system coefficients
```

```
> coeff est = Hpinv#(s+n)
  print, 'RMS: ', sqrt(mean(abs(coeff_est - coeff)^2))
 ;; Fitted signal
  s_fit = H#coeff_est
> plot, x, s+n, linestyle=1, thick=2.; Noisy observation
                           ; True signal
> oplot, x, s, color='ff'x
> oplot, x, s fit, color='ff00'x
                               ; Fitted signal
> 8<-----
> If your sample vector X happens to be regular, the solution to your problem is actually nothing
more than an FFT, and pick the 5 first complex coefficients. The first coefficient is the constant
term A0 (not included in your problem), and real/imaginary parts of the following coefficients
corresponds to cosine terms A1-A4 and the sine terms B1-B4, respectively.
>
> Yngvar
>
> On Saturday, 25 October 2014 07:47:17 UTC+2, siumt...@gmail.com wrote:
   I think you should try to be for specific to ask question here.
>>
>> Suppose I have a timeseries with the S size.
>> I want to do nonlinear fitting to the timeseries using the following fourier series (harmonic
function)
>>
>>
>> And I would find 8 coefficients such as An and Bn where n = 1,2,3,4
>>
   That is.
>>
>> A1,A2,A3,A4
>> B1,B2,B3,B4
>>
>> I have attempted to understand how it works mpfit by Craig and curvefit. Unfortunately, I did
not because I am not IDL expert. So I posted this if anyone can help
>>
>>
>> Best Wishes
>>
>>
>> Thanks for you help
```

Thanks for your help

I have attempted to used multiple linear regression to solve the problem. However, when I plot the original data with fitted line, I did not find good result.

I think it is better i put sampledata which I understand instead of random numbers

Here is sample data (sampledata.txt) 8.82609 8.50000 7.52174 6.20833 8.11111 10.4000 8.39286 9.19231 7.30769 9.13043 7.57692 8.00000 10.9600 8.33333 8.90476 10.8750 10.6250 10.2069 10.0000 11.7391 10.6538 11.8500 8.68000 8.96296 6.72727 7.56522 7.47826 7.96296 9.03846 9.19231 8.81818 7.08333 8.69565 8.75000 8.55556 8.28000 8.68000 7.87500 7.71429

7.78261 6.85714

- 9.11111
- 6.46154
- 6.92593
- 6.77778
- 6.50000
- 7.38889
- 8.81818
- 5.45455
- 4.72727
- 5.04348
- 5.75000
- 6.60870
- 5.91304 6.71429
- 7.33333
- 7.21739
- 6.54545
- 6.62500
- 5.94737
- 6.72000
- 7.28000
- 7.82353
- 7.50000 8.80952
- 9.59091
- 8.04348
- 7.44444
- 6.80000
- 8.55556
- 12.5556
- 6.47368
- 7.41176
- 7.80000
- 7.68750
- 7.31579
- 7.31579
- 8.37500
- 7.00000
- 9.88235
- 9.42105
- 9.41177
- 7.53333
- 6.70588
- 9.30000
- 8.66667
- 9.26316
- 11.4091
- 7.35000

- 9.16667
- 8.09091
- 8.84210
- 7.05000
- 8.85000
- 8.32000
- 8.66667
- 7.25000
- 6.77778
- 6.15000
- 8.70588
- 7.52381
- 7.91304 7.33333
- 6.64706
- 7.26316
- 7.95238
- 7.00000
- 7.45000
- 7.30769
- 11.0000
- 12.8750
- 8.18182
- 11.8182
- 9.09091
- 10.3750
- 11.0833
- 11.7333
- 12.7857
- 11.1667
- 12.5833
- 7.76190
- 6.06667
- 6.28571
- 6.00000
- 9.76190
- 8.62500
- 6.31579
- 6.00000
- 9.20000
- 9.47059
- 9.42857
- 8.05882
- 9.56522
- 9.30435
- 9.80000
- 8.42857
- 8.65000

- 8.44444
- 8.56522
- 6.83333
- 9.52381
- 9.11539
- 8.00000
- 6.42857
- 9.36364
- 8.80000
- 9.02308
- 7.90909
- 7.95455
- 8.54091
- 8.55556
- 7.95652
- 10.8621
- 5.80000
- 8.84167
- 8.34615
- 7.16000
- 6.47619
- 6.95238
- 6.76190
- 6.48571
- 9.95833
- 7.65217
- 6.86500
- 8.16000
- 7.57619
- 7.68182
- 7.04762
- 8.77727
- 7.82917
- 8.96923
- 7.52174
- 8.82083
- 7.59583
- 8.23810
- 8.60476
- 9.22609
- 7.86522
- 10.1083
- 8.26667
- 8.68095
- 10.4632
- 8.23333
- 7.85217
- 6.85455

- 7.24444
- 7.56522
- 7.28947
- 7.33000
- 8.97369
- 7.84500
- 9.40000
- 5.55882
- 7.30870
- 6.39500
- 8.12000
- 8.42222
- 7.87917 6.90625
- 7.46000
- 6.22632
- 7.98667
- 7.96842
- 6.99474
- 10.9944
- 8.29474
- 8.40417
- 7.93637
- 8.67727
- 7.49444
- 7.61923
- 8.04000
- 8.12857
- 9.06667
- 9.45500
- 10.1833 10.6923
- 11.3458
- 12.0952
- 10.7346
- 10.3429
- 9.97391
- 8.39091
- 7.49565
- 8.06538
- 8.71200
- 8.90952
- 9.08750
- 9.18077
- 10.5808
- 8.29524
- 9.22727
- 8.22857

- 7.56500
- 7.26087
- 7.86667
- 10.1913
- 9.27693
- 7.83044
- 7.21818
- 7.41053
- 7.91923
- 8.43077
- 8.37500
- 9.40417
- 7.23684
- 6.79444
- 8.24348
- 7.19259
- 7.90400
- 8.15000
- 7.40000
- 6.35455 7.39500
- 8.88261
- 7.66522
- 9.07143
- 8.88750
- 7.27917
- 8.92917
- 8.28333
- 7.74231
- 7.47600
- 6.93182
- 7.92609
- 8.21379
- 7.89286
- 7.80000
- 6.84800
- 7.63200
- 7.78333
- 7.65600
- 7.46786
- 7.10000
- 7.59600
- 7.13750
- 9.67000
- 13.1400
- 12.3037
- 11.5038
- 11.5087

- 10.2235
- 8.43158
- 8.12273
- 40.0200
- 10.2700
- 8.30417
- 10.5952
- 9.98800
- 7.63200
- 8.22692
- 0.22032
- 9.49167
- 7.63333
- 7.48333
- 9.22917
- 9.34643
- 7.10000
- 8.17692
- 7.15417
- 5.82174
- 8.90800
- 8.01600
- 7.42381
- 8.43333
- 9.04074
- 8.56923
- 8.25200
- 0.20200
- 8.60741
- 8.19655
- 11.4400
- 7.57500
- 8.35000
- 9.83478
- 10.4069
- 11.1704
- 12.8808
- 10.3091
- 13.2364
- 10.8391
- 10.5560
- 10.4704
- 11.1767
- 9.77143
- 11.1520
- 10.9652
- 9.19643
- 9.11539
- 11.3667
- 11.0444
- 10.6370

- 10.8714
- 9.14231
- 10.4385
- 11.0593
- 10.5760
- 10.5700
- 9.15000
- 10.0231
- 8.59615
- 9.68461
- 9.26000
- 7.51852
- 7.21923
- 1.21323
- 8.53333
- 10.0133
- 9.27857
- 7.65000
- 9.12143
- 10.1560
- 7.86400
- 8.10385
- 7.89231
- 10.0600
- 10.0000
- 8.06667
- 7.19643
- 6.87308
- 6.98800
- 7.14074
- 6.91429
- 8.03333
- 7.32174
- 7.67200
- 9.14444
- 8.24400
- 10.1148
- 11.3731
- 10.5138
- 9.80000
- 9.10800
- 7.65556
- 7.36667
- 7.42500
- 6.92308
- 7.96923
- 8.80000
- 10.3250
- 9.53929
- 9.77308
- 9.78929

10.2077 9.06154 8.71538 8.67143 9.41923 8.90000 9.44400 7.64231 9.06000 8.14445 9.87917 9.49167 9.74615 8.74231 9.73214 10.1400 9.63214 9.23462 8.83043 8.54615 8.53333 7.62308 8.65357 9.45000 9.02759 8.82857 8.54231 9.37857 9.84815 10.6966

file='sampledata.txt' readcol,file,input

10.3143

pro funcname,input=input,scycle=scycle,X1=X1,X9=X9

n=n\_elements(input) scycle=fltarr(n)+1E20 X1=fltarr(n)+1E20 X2=fltarr(n)+1E20 X3=fltarr(n)+1E20 X4=fltarr(n)+1E20 X5=fltarr(n)+1E20 X6=fltarr(n)+1E20 X7=fltarr(n)+1E20

```
X8=fltarr(n)+1E20
X9=fltarr(n)+1E20
tmpX9=(findgen(n)+1)/12.
t=findgen(n)+1
dependvar=fltarr(n)
pi = !PI*1.0
clean=where(input LT 1000 and input NE 0)
if clean(0) GE 0 and n_elements(clean) GE 0.9*n_elements(clean) then begin
for i=0,n_elements(clean)-1 do begin
 g=clean(i)
 dependvar(g)=input(g)
   angle=(2*pi*t(g))/12.
   X1(g)=\sin(angle)
   X2(g)=cos(angle)
   X3(g)=\sin(2*angle)
   X4(g)=cos(2*angle)
   X5(g)=\sin(3*angle)
   X6(g)=\cos(3*angle)
   X7(g)=\sin(4*angle)
   X8(q)=\cos(4*angle)
   X9(q)=tmpX9(q)
endfor
X = [TRANSPOSE(X1), TRANSPOSE(X2), TRANSPOSE(X3),$
 TRANSPOSE(X4), TRANSPOSE(X5), TRANSPOSE(X6),$
 TRANSPOSE(X7), TRANSPOSE(X8), TRANSPOSE(X9)]
 weights = REPLICATE(1.0, n); Create an Npoints-element vector of uniform weights.
 coeffs=regress(X(*,clean),dependvar(clean),weights(clean),SI GMA=sigma, CONST=const, $
 MEASURE_ERRORS=measure_errors,yfit=yfit, /RELATIVE_WEIGHT)
print, 'multiple linear regression coeffs'
print, coeffs
```

```
print,'slope'
print,coeffs[8]
print,yfit
constv=fltarr(n)
constv(*)=const
FOR k=0,n_elements(clean)-1 do begin
 j=clean(k)
  scycle(j) = coeffs(0)*X1(j)+coeffs(1)*X2(j)+$
   coeffs(2)*X3(j)+coeffs(3)*X4(j)+$
   coeffs(4)*X5(j)+coeffs(5)*X6(j)+$
   coeffs(6)*X7(j)+coeffs(7)*X8(j)+constv(j)+coeffs(8)*X9(j)
 ENDFOR; i loop
endif else begin
 dependvar(*)=1E20
   X1(*)=1E20
   X2(*)=1E20
   X3(*)=1E20
   X4(*)=1E20
   X5(*)=1E20
   X6(*)=1E20
   X7(*)=1E20
   X8(*)=1E20
   X9(*)=1E20
 scycle(*)=1E20
 endelse
;print,X1
print, 'number of X1'
print,n_elements(X1)
print,const
END
```

Subject: Re: Curve Fitting to timeseries using a set of 8 sine and cosine functions Posted by siumtesfai on Sat, 25 Oct 2014 17:51:02 GMT View Forum Message <> Reply to Message

Thank You for spending some time to help me. ..

Sorry If I post the same question again and again. I just need a solution for my problem.

I have attempted to use multiple linear regression to solve the problem. However, when I plot the

original data with fitted value, I did not find good result.

I think it is better I put sampedata which I understand instead of using random numbers

Here is sample data (sampledata.txt) 8.82609 8.50000 7.52174 6.20833 8.11111 10.4000 8.39286 9.19231 7.30769 9.13043 7.57692 8.00000 10.9600 8.33333 8.90476 10.8750 10.6250 10.2069 10.0000 11.7391 10.6538 11.8500 8.68000 8.96296 6.72727 7.56522 7.47826 7.96296 9.03846 9.19231 8.81818 7.08333 8.69565 8.75000 8.55556 8.28000 8.68000 7.87500 7.71429

7.78261 6.85714 9.11111

- 6.46154
- 6.92593
- 6.77778
- 6.50000
- 7.38889
- 8.81818
- 5.45455
- 4.72727
- 5.04348
- 5.75000
- 6.60870
- 5.91304
- 6.71429
- 7.33333
- 7.21739
- 6.54545
- 6.62500
- 5.94737
- 6.72000
- 7.28000 7.82353
- 7.50000
- 8.80952
- 9.59091
- 8.04348
- 7.44444
- 6.80000
- 8.55556
- 12.5556
- 6.47368
- 7.41176
- 7.80000
- 7.68750
- 7.31579
- 7.31579
- 8.37500
- 7.00000
- 9.88235
- 9.42105
- 9.41177
- 7.53333
- 6.70588
- 9.30000
- 8.66667
- 9.26316
- 11.4091
- 7.35000
- 9.16667

- 8.09091
- 8.84210
- 7.05000
- 8.85000
- 8.32000
- 8.66667
- 7.25000
- 6.77778
- 6.15000
- 8.70588
- 7.52381
- 7.91304
- 7.33333
- 6.64706
- 7.26316
- 7.95238
- 7.00000
- 7.45000
- 7.30769
- 11.0000 12.8750
- 8.18182 11.8182
- 9.09091
- 10.3750
- 11.0833
- 11.7333
- 12.7857
- 11.1667
- 12.5833 7.76190
- 6.06667
- 6.28571
- 6.00000
- 9.76190
- 8.62500
- 6.31579
- 6.00000
- 9.20000
- 9.47059
- 9.42857
- 8.05882
- 9.56522
- 9.30435
- 9.80000
- 8.42857
- 8.65000
- 8.44444

- 8.56522
- 6.83333
- 9.52381
- 9.11539
- 5.11555
- 8.00000
- 6.42857
- 9.36364
- 8.80000
- 9.02308
- 7.90909
- 7.95455
- 8.54091
- 8.55556
- 7.95652
- 10.8621
- 5.80000
- 8.84167
- 8.34615
- 7.16000
- 6.47619
- 6.95238
- 6.76190
- 6.48571
- 9.95833
- 7.65217
- 7.00217
- 6.86500
- 8.16000
- 7.57619
- 7.68182
- 7.04762
- 8.77727
- 7.82917
- 8.96923
- 7.52174
- 8.82083
- 7.59583
- 8.23810
- 8.60476
- 9.22609
- 7.86522
- 10.1083
- 8.26667
- 8.68095
- 10.4632
- 8.23333
- 7.85217
- 6.85455
- 7.24444

- 7.56522
- 7.28947
- 7.33000
- 8.97369
- 7.84500
- 9.40000
- 5.55882
- 7.30870
- 6.39500
- 8.12000
- 8.42222
- 7.87917
- 6.90625
- 7.46000
- 6.22632
- 7.98667
- 7.96842
- 6.99474
- 10.9944
- 8.29474
- 8.40417
- 7.93637
- 8.67727
- 7.49444
- 7.61923
- 8.04000
- 8.12857
- 9.06667
- 9.45500
- 10.1833
- 10.6923
- 11.3458
- 12.0952
- 10.7346
- 10.3429
- 9.97391
- 8.39091
- 7.49565
- 8.06538
- 8.71200
- 8.90952
- 9.08750
- 9.18077
- 10.5808
- 8.29524
- 9.22727
- 8.22857
- 7.56500

- 7.26087
- 7.86667
- 10.1913
- 9.27693
- 7.83044
- 7.21818
- 7.41053
- 7.91923
- 8.43077
- 8.37500
- 9.40417
- 7.23684
- 6.79444
- 8.24348
- 7.19259
- 7.90400
- 8.15000
- 7.40000
- 6.35455
- 7.39500
- 8.88261
- 7.66522
- 9.07143
- 8.88750
- 7.27917
- 8.92917
- 8.28333
- 7.74231
- 7.47600
- 6.93182
- 7.92609
- 8.21379
- 7.89286
- 7.80000
- 6.84800
- 7.63200
- 7.78333
- 7.65600
- 7.46786
- 7.10000
- 7.59600
- 7.13750
- 9.67000
- 13.1400
- 12.3037
- 11.5038
- 11.5087
- 10.2235

- 8.43158
- 8.12273
- 10.2700
- 8.30417
- 10.5952
- 9.98800
- 7.63200
- 8.22692
- 9.49167
- 7.63333
- 7.48333
- 9.22917
- 9.34643
- 7.10000
- 8.17692
- 7.15417
- 5.82174
- 8.90800
- 8.01600
- 7.42381
- 8.43333
- 9.04074
- 8.56923
- 8.25200
- 8.60741
- 8.19655
- 11.4400
- 7.57500
- 8.35000
- 9.83478
- 10.4069
- 11.1704
- 12.8808
- 10.3091
- 13.2364
- 10.8391
- 10.5560
- 10.4704
- 11.1767
- 9.77143
- 11.1520
- 10.9652 9.19643
- 9.11539
- 11.3667
- 11.0444 10.6370
- 10.8714

- 9.14231
- 10.4385
- 11.0593
- 10.5760
- 9.15000
- 10.0231
- 8.59615
- 9.68461
- 9.26000
- 7.51852
- 7.21923
- 8.53333
- 10.0133
- 9.27857
- 7.65000
- 9.12143
- 10.1560
- 7.86400
- 8.10385
- 7.89231
- 10.0600
- 8.06667
- 7.19643
- 6.87308
- 6.98800
- 7.14074
- 6.91429
- 8.03333
- 7.32174
- 7.67200
- 9.14444
- 8.24400
- 10.1148
- 11.3731
- 10.5138
- 9.80000
- 9.10800
- 7.65556
- 7.36667
- 7.42500
- 6.92308
- 7.96923
- 8.80000
- 10.3250
- 9.53929
- 9.77308
- 9.78929
- 10.2077

9.06154

8.71538

8.67143

9.41923

8.90000

9.44400

3.44400

7.64231

9.06000

8.14445

9.87917

9.49167

9.74615

8.74231

9.73214

10.1400

9.63214

9.23462

8.83043

8.54615

8.53333

7.62308

8.65357

9.45000

9.02759

8.82857

8.54231

9.37857

9.84815

10.6966

10.3143

file='sampledata.txt' readcol,file,input

pro funcname,input=input,scycle=scycle,X1=X1,X9=X9

n=n\_elements(input)

scycle=fltarr(n)+1E20

X1=fltarr(n)+1E20

X2=fltarr(n)+1E20

X3=fltarr(n)+1E20

X4=fltarr(n)+1E20

X5=fltarr(n)+1E20

X6=fltarr(n)+1E20

X7=fltarr(n)+1E20

X8=fltarr(n)+1E20

```
X9=fltarr(n)+1E20
tmpX9=(findgen(n)+1)/12.
t=findgen(n)+1
dependvar=fltarr(n)
pi=!PI*1.0
clean=where(input LT 1000 and input NE 0)
if clean(0) GE 0 and n elements(clean) GE 0.9*n elements(clean) then begin
for i=0,n elements(clean)-1 do begin
 g=clean(i)
 dependvar(g)=input(g)
   angle=(2*pi*t(g))/12.
   X1(g)=\sin(angle)
   X2(g)=cos(angle)
   X3(q)=\sin(2^*angle)
   X4(g)=\cos(2^*angle)
   X5(g)=\sin(3*angle)
   X6(g)=cos(3*angle)
   X7(g)=\sin(4*angle)
   X8(g)=cos(4*angle)
   X9(g)=tmpX9(g)
endfor
X = [TRANSPOSE(X1), TRANSPOSE(X2), TRANSPOSE(X3),$
 TRANSPOSE(X4), TRANSPOSE(X5), TRANSPOSE(X6),$
 TRANSPOSE(X7), TRANSPOSE(X8), TRANSPOSE(X9)]
 weights = REPLICATE(1.0, n); Create an Npoints-element vector of uniform weights.
 coeffs=regress(X(*,clean),dependvar(clean),weights(clean),SI GMA=sigma, CONST=const, $
 MEASURE_ERRORS=measure_errors,yfit=yfit, /RELATIVE_WEIGHT)
print, 'multiple linear regression coeffs'
print, coeffs
print, 'slope'
```

```
print,coeffs[8]
print,yfit
constv=fltarr(n)
constv(*)=const
FOR k=0,n_elements(clean)-1 do begin
 j=clean(k)
  scycle(j) = coeffs(0)*X1(j)+coeffs(1)*X2(j)+$
   coeffs(2)*X3(j)+coeffs(3)*X4(j)+$
   coeffs(4)*X5(j)+coeffs(5)*X6(j)+$
   coeffs(6)*X7(j)+coeffs(7)*X8(j)+constv(j)+coeffs(8)*X9(j)
 ENDFOR; j loop
endif else begin
 dependvar(*)=1E20
   X1(*)=1E20
   X2(*)=1E20
   X3(*)=1E20
   X4(*)=1E20
   X5(*)=1E20
   X6(*)=1E20
   X7(*)=1E20
   X8(*)=1E20
   X9(*)=1E20
 scycle(*)=1E20
 endelse
;print,X1
print, 'number of X1'
print,n_elements(X1)
print,const
;==============
  plot to check the result
xaxis=findgen(n_elements(input))
cgplot,xaxis, input ,color =cgcolor('black'); orignial data
oplot,xaxis, scycle,color = cgcolor('blue') ; fitted
END
```

Subject: Re: Curve Fitting to timeseries using a set of 8 sine and cosine functions Posted by Yngvar Larsen on Sat, 25 Oct 2014 19:34:39 GMT

View Forum Message <> Reply to Message

>

On Saturday, 25 October 2014 19:51:05 UTC+2, siumt...@gmail.com wrote:

- > Thank You for spending some time to help me. ..
- > Sorry If I post the same question again and again. I just need a solution for my problem.
- > I have attempted to use multiple linear regression to solve the problem . However, when I plot the original data with fitted value , I did not find good result.
- > I think it is better I put sampedata which I understand instead of using random numbers

[Note that I did not use completely random numbers in my example, just arbitrary coefficients in the model.]

You claim you understand your sample data. But you should really explore your data model a bit more. You are basically modelling your data as periodic, with only the frequencies 1-4/yr. The following spectral analysis shows that your signal in fact does not contain much energy at these frequencies (exact maybe the seasonal signal at 1/yr). Vertical green lines correspond to your model frequencies, and the red ones are the major ones found by a simple spectral analysis.

Bottom line: your model is very wrong, so no wonder estimating the model coefficients using linear regression/least squares does not work well.

```
8<-----
:: read test data
datafile = 'sampledata.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
;; one-sided periodogram
pow = (abs(fft(data*hanning(np)))^2)[1:np/2]
```

```
plot, faxis, pow, xtitle='Frequency [1/years]', ytitle='Power spectrum', /xlog
;; Dominant modes (eyeball fit)
oplot, df*[1,1]*1.1, !y.crange, color='ff'x
oplot, df*[1,1]*4, !y.crange, color='ff'x
oplot, df*[1,1]*8, !y.crange, color='ff'x
oplot, df*[1,1]*12, !y.crange, color='ff'x
oplot, df*[1,1]*15, !y.crange, color='ff'x
oplot, df*[1,1]*19.5, !y.crange, color='ff'x
oplot, df*[1,1]*25, !y.crange, color='ff'x
oplot, df*[1,1]*28, !y.crange, color='ff'x
oplot, df*[1,1]*33, !y.crange, color='ff'x
oplot, df*[1,1]*41, !y.crange, color='ff'x
oplot, df*[1,1]*55, !y.crange, color='ff'x
oplot, df*[1,1]*81, !y.crange, color='ff'x
oplot, df*[1,1]*124, !y.crange, color='ff'x
oplot, df*[1,1]*164, !y.crange, color='ff'x
oplot, df*[1,1]*192, !y.crange, color='ff'x
;; Your assumed modes
for n=1,4 do $
  oplot, [1,1]*n, !y.crange, color='ff00'x
end
8<-----
Yngvar
```

Subject: Re: Curve Fitting to timeseries using a set of 8 sine and cosine functions Posted by siumtesfai on Sat, 25 Oct 2014 23:00:07 GMT

View Forum Message <> Reply to Message

>

On Saturday, October 25, 2014 1:47:17 AM UTC-4, siumt...@gmail.com wrote:

> I think you should try to be for specific to ask question here.

> Suppose I have a timeseries with the S size.

> I want to do nonlinear fitting to the timeseries using the following fourier series ( harmonic function)

> And I would find 8 coefficients such as An and Bn where n = 1,2,3,4

> That is. > A1,A2,A3,A4 > B1,B2,B3,B4 >

> I have attempted to understand how it works mpfit by Craig and curvefit. Unfortunately, I did not because I am not IDL expert. So I posted this if anyone can help

>

> Best Wishes

*-*

> Thanks for you help

Hello,

I do not use FFT because I have missing data . I provided you with monthly timeseries which does not have missing data. But generally, I use monthly datasets that have missing values.

That is why I used multiple regression.

Assumption about regression is though that the dataset follow gaussian distribution. Maybe I am not getting the correct coefficients of the sine and cosine terms because my data is skewed (I can not think any thing else)

Second, I believe my model is correct because Suppose you have monthly temperature datasets. You can represent your seasonal cycle using the harmonics as I have .

**Thanks** 

Subject: Re: Curve Fitting to timeseries using a set of 8 sine and cosine functions Posted by Yngvar Larsen on Sun, 26 Oct 2014 18:05:22 GMT View Forum Message <> Reply to Message

On Sunday, 26 October 2014 01:00:08 UTC+2, siumt...@gmail.com wrote:

> I do not use FFT because I have missing data . I provided you with monthly timeseries which does not have missing data. But generally, I use monthly datasets that have missing values.

Ok. You should have told so then...

However, you can still repeat my harmonic analysis using a Fourier transform for irregularly sampled data (which is really a kind of multiple regression since the fourier transform is linear).

> That is why I used multiple regression.

>

> Assumption about regression is though that the dataset follow gaussian distribution. Maybe I am not getting the correct coefficients of the sine and cosine terms because my data is skewed (I can not think any thing else)

This is true in some sense, since there residual signal after subtracting your seasonal signal still is significant and not really noiselike (nonzero skewness and/or kurtosis, colored). One could say that you have a very low SNR, if you interpret the residual signal as noise.

> Second, I believe my model is correct because Suppose you have monthly temperature datasets. You can represent your seasonal cycle using the harmonics as I have .

You haven't told whether your goal is to extract the seasonal signal or to fit the full signal to some mode.

Indeed you can respresent a lowpass seasonal signal with period 12 moths with your model, using 2x4 coefficents (plus a constant term you should include or subtract from the original signal). What I meant is that at least for the signal you posted, this seasonal component does not contain much energy. Thus, your model yields a very bad fit to your original signal. I showed in my previous post, with a simplified harmonic analysis, that most of your signal energy is contained in harmonic components with longer periods than 12 months.

However, this is now diverging away from IDL and into signal processing theory. Your problem is not IDL. From the style of your code, I assume you inherited it from someone who wrote it in the 90's (or didn't change style since then...). The code probably works correctly for the problem it was written for. REGRESS has a few keywords that will provide goodness-of-fit statistics. I suggest you try them, and you will see that your model is not good for the full signal.

Yngvar

Subject: Re: Curve Fitting to timeseries using a set of 8 sine and cosine functions Posted by siumtesfai on Mon, 27 Oct 2014 01:45:48 GMT View Forum Message <> Reply to Message

Thanks for your discussion.

oh also for your help

One correction

I am not developing a code that is written in 90'S. Actually, the models is widely used. I can provide you a paper about the model.

I wonder that the models works for others using similar dataset but not for me. That is why I posted the question. Maybe there is better way dealing with the problem.

Subject: Re: Curve Fitting to timeseries using a set of 8 sine and cosine functions Posted by Yngvar Larsen on Mon, 27 Oct 2014 13:03:03 GMT

View Forum Message <> Reply to Message

On Monday, 27 October 2014 02:45:50 UTC+1, siumt...@gmail.com wrote:

- > Thanks for your discussion.
- >
- > oh also for your help

>

> One correction

>

> I am not developing a code that is written in 90'S. Actually, the models is widely used. I can provide you a paper about the model.

I'm not talking about the algorithm, which probably works fine for what it was intended for. I'm talking about the coding style.

- 1) Round parens array(i) for array indexing was deprecated in favor of array[i] around 1997. Don't expect this to work forever!
- 2) Uppercase GT/LT/LE/etc and keywords is typical for people using old editors without syntax highlighting. Syntax highlighting should be supported by any decent editor used today. This one is more a matter of taste though.
- > I wonder that the models works for others using similar dataset but not for me. That is why I posted the question. Maybe there is better way dealing with the problem.

You still haven't told what you are trying to do or what your signal really is?

\* Extract seasonal component? It turns out to be almost not present in the signal you posted.

or

\*Fitting a harmonic series to your signal? You need to use more terms, with longer periods then 12 monts.

Yngvar

Subject: Re: Curve Fitting to timeseries using a set of 8 sine and cosine functions Posted by Craig Markwardt on Mon, 27 Oct 2014 15:52:33 GMT View Forum Message <> Reply to Message

On Saturday, October 25, 2014 7:00:08 PM UTC-4, siumt...@gmail.com wrote:

> I do not use FFT because I have missing data . I provided you with monthly timeseries which does not have missing data. But generally, I use monthly datasets that have missing values.

>

> That is why I used multiple regression.

You can still use an FFT for regularly sampled data, but with some missing points. Just replace the missing points with zero. A sample value of 0 does not contribute power to a Fourier transform. (this is why zero-padding works)

Actually, it's better to first subtract the mean value of the time series (ignoring missing values), then replace missing values with zero. This minimizes the chances of an alias of the DC term from getting into your data.

Craig

Subject: Re: Curve Fitting to timeseries using a set of 8 sine and cosine functions Posted by siumtesfai on Tue, 28 Oct 2014 02:16:05 GMT

View Forum Message <> Reply to Message

On Monday, October 27, 2014 11:52:36 AM UTC-4, Craig Markwardt wrote:

- > On Saturday, October 25, 2014 7:00:08 PM UTC-4, siumt...@gmail.com wrote:
- >> I do not use FFT because I have missing data . I provided you with monthly timeseries which does not have missing data. But generally, I use monthly datasets that have missing values.

>>

>> That is why I used multiple regression.

>

> You can still use an FFT for regularly sampled data, but with some missing points. Just replace the missing points with zero. A sample value of 0 does not contribute power to a Fourier transform. (this is why zero-padding works)

>

> Actually, it's better to first subtract the mean value of the time series (ignoring missing values), then replace missing values with zero. This minimizes the chances of an alias of the DC term from getting into your data.

> Craig

thanks both

I will trying to extract seasonal cycle from my timeseries. Eventhoug I do not have seasonal cycle for the data I provided, it does not mean that there is no seasonal cycle. Please consider I have temperature timeseries which shows cold winter and warm summer seasonal cycle. So I should be able to represent my seasonal cycle using the four sines and cosine function (harmonics). All I am asking is why is it that I do not get the seasonal cycle coefficients right when I use multiple regression. Let me know if I am becoming stagnant with the idea that I have to change to fourier series analysis.

Best regards,

Sorry guys if I do not understand you

Subject: Re: Curve Fitting to timeseries using a set of 8 sine and cosine functions Posted by siumtesfai on Sat, 01 Nov 2014 03:12:00 GMT

View Forum Message <> Reply to Message

On Saturday, October 25, 2014 1:47:17 AM UTC-4, siumt...@gmail.com wrote:

> I think you should try to be for specific to ask question here.

> Suppose I have a timeseries with the S size.

> I want to do nonlinear fitting to the timeseries using the following fourier series ( harmonic function)

> And I would find 8 coefficients such as An and Bn where n = 1,2,3,4

> That is.

> A1,A2,A3,A4

> B1,B2,B3,B4

> I have attempted to understand how it works mpfit by Craig and curvefit . Unfortunately, I did not because I am not IDL expert. So I posted this if anyone can help

> Best Wishes

> Thanks for you help

Hello again Yngyar and craig

Hello again Yngvar and craig

My question is not resolved yet.

Craig: Why you did not suggest me to use your MPFIT code for fitting my model.

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.

**Thanks** 

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
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
```

Subject: Re: Curve Fitting to timeseries using a set of 8 sine and cosine functions Posted by Yngvar Larsen on Sat, 01 Nov 2014 19:05:28 GMT

View Forum Message <> Reply to Message

On Saturday, 1 November 2014 15:57:26 UTC+1, Yngvar Larsen wrote:

> 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:)

Jeez. I'm not winning any spelling contests today, that's for sure... Goodness-of-pit??

--

Yngvar

Subject: Re: Curve Fitting to timeseries using a set of 8 sine and cosine functions Posted by Craig Markwardt on Mon, 03 Nov 2014 04:49:08 GMT View Forum Message <> Reply to Message

On Friday, October 31, 2014 11:12:03 PM UTC-4, 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.

You can use MPFIT if you want. As you mentioned in the original post, you knew about MPFIT already, so why did I need to suggest it again?

In any case, since an FFT does produce a least squares set of fourier coefficients, using MPFIT is kind of overkill.

You said this,

- > I have attempted to used multiple linear regression to solve the problem .
- > However, when I plot the original data with fitted line, I did not find good
- > result.

So - given your model - how do you know there is a better result?

Craig

Subject: Re: Curve Fitting to timeseries using a set of 8 sine and cosine functions Posted by siumtesfai on Tue, 04 Nov 2014 23:14:59 GMT

View Forum Message <> Reply to Message

Thanks both,

You helped me.

Best regards

On Sunday, November 2, 2014 11:49:10 PM UTC-5, Craig Markwardt wrote:

- > On Friday, October 31, 2014 11:12:03 PM UTC-4, 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.

```
> You can use MPFIT if you want. As you mentioned in the original post, you knew about MPFIT already, so why did I need to suggest it again?
> In any case, since an FFT does produce a least squares set of fourier coefficients, using MPFIT is kind of overkill.
> You said this,
> You said this,
> However, when I plot the original data with fitted line, I did not find good
> result.
> So - given your model - how do you know there is a better result?
```

> Craig