Subject: EOF Analysis
Posted by B}rd Krane on Fri, 09 May 1997 07:00:00 GMT
View Forum Message <> Reply to Message

Recently I received an email Julian Castaneda with questions about
Empirical Orthogonal Functions. Unfortunately the reply address was
bogus and my reply just bounced. I am quite sure my name was obtained
from one of my postings in this newsgroup and as a last try I post my
reply here.

```
;Dear Julian
;I am still fairly new to Singular Value Analysis, or EOF, but
;perhaps these references could be of interest:
;
; Roberts, R. A and Mullis, C. T.
; "Digital Signal Processing",
; 1987 Addison-Wesley, pp 538-542
; ISBN: 0-201-16350-0
;
; Johnson, D. E. and Dudgeon, D. E.
; "Array Signal Processing; Concepts and Techniques",
; 1993 Prentice-Hall, 496-497
; ISBN: 0-13-048513-6
;
;I guess you obtained my name from the IDL-PVWAVE newsgroup and I
;therefore include a little code fragment in IDL. If you prefer other
;programming languages I would recommend the Numerical Recipes since
;the IDL-code is based on this book.
;
;B{\aa}rd Krane                | Email : bard.krane@fys.uio.no
;Institute of Physics          | Phone : + 47 22 85 56 66
;University of Oslo            | Fax   : + 47 22 85 56 71
;POB 1048, blindern, N-0316 Norway |

; --- IDL Code ---
N = 32                                  ; Spatial Resolution
M = 64                                  ; Temporal Resolution

x = 2.0*!pi*findgen(N)/N                    ; 0 <= x < 2pi
t = findgen(M)/M                          ; 0 <= t < 1.0

x = x # (fltarr(M)+1.0)                     ; Create a NxM  matrix
                                        ; with an outer product
t = (fltarr(N)+1.0) # t                     ; This one is also NxM

; Try this out for different Signal to Noise ratios and
; see the effect on the distribution of Singular Values
A = sin(x)*sin(2.5*!pi*t) + 0.5*randomu(seed,N,M) ; Standing Wave +
```

```
Noise
A = sin(x+2.5*!pi*t) + 0.5*randomu(seed,N,M)    ; Propagating Wave +
Noise

svdc,A,W,U,V                            ; Singular Value
Decomposition
                                  ; Check the IDL Manual

idx = reverse(sort(W))                    ; Sort the singular
values
W = W(idx)                              ; according to size
V = V(idx,*)
U = U(idx,*)

K = n_elements(where(W GE 0.1*W(0)))          ; Decide how many terms
to
                                  ; use in the
reconstruction
B = fltarr(N,M)
FOR i=0,K-1 DO $                          ; Reconstruct with K
terms
   B = B + W(i) * reform(V(i,*)) # reform(U(i,*))


window,1,retain=2
!p.multi = [0,3,1]
!x.style = 1
!y.style = 1

contour,A,nlevels=12,title="Sampled Data"
contour,B,nlevels=12,title="Reconstructed Data"
plot,w/w(0),yrange=[0.01,2.0],/ylog,psym=2,title="Singular Values"

end
```