

Coding of covariance data in EXFOR convenient for software

V.Zerkin, IAEA-NDS, 20/05/2011

“Technical Meeting Neutron Cross Section Covariances, IAEA Headquarters Vienna, Austria 27-30 September 2010”:

- The EXFOR formats should be flexible enough to accommodate information as provided by the experimenters. The EXFOR format should be extended to accommodate the full explicit covariance matrix, if provided by the authors or, alternatively, data required by a recipe for constructing covariance matrices based on providing partial error and correlation components.

Today EXFOR has very few entries having some information about experimental covariance data: 43 Entry with keyword COVARIANCE having data as free text and 6 Entries with code COVAR (files *.COV with covariance data in ENDF format have been lost). Because there are no coding rules for covariance data placed in EXFOR files as free text, but data exist, and EXFOR users are interested to have such a data in the EXFOR files, there is a need to introduce rules (conventions) for coding covariance data. This paper is presented to NRDC community in order to emphasise that the coding rule should be not overcomplicated for users, which means nowadays – not too difficult for programming.

Proposal:

1. to introduce a convention to present existing in EXFOR free text covariance data in a unified style simple for programming (possibly compatible with z(x,y) input format of plotting package ZVView, used from 2009);
2. to foresee extension of the coding rules for the needs/convenience of EXFOR compilers/physicists.

The essential and absolutely necessary information for program reading covariance matrix Z(X,Y):

- X array (dimension, units, name of axis)
- Y array (dimension, units, name of axis)
- Z array (dimension, units, name of axis)

Typical options:

- X array can be equal to Y array (e.g. squared matrix energy-energy correlation)
- Z array can be given as triangle (symmetric squared matrix)
- matrix can present various correlations (e.g. reaction-reaction)

The proposed format is following:

- For every array:
 - a. Line-Header for every array:
code in brackets starting from 12 column;
first values are fixed: type of array (X, Y, Z, XY), dimension (integer), meaning (e.g. EN) and units (e.g. MeV); the rest can be extended later
 - b. Following lines with real numbers separated by blanks until number of values reaches dimension of array

Examples:

1) Triangle symmetric matrix Energy-Energy correlation (EXFOR: 2211003)

123456789.123456789.123456789.123456789.123456789.123456789.123456

COVARIANCE .Covariation matrix in percent

(XY,17,EN,MEV)

0.597 0.797 0.898 0.906 1.01 1.098 1.117 1.193 1.376
1.39 1.693 2.144 3.908 5.182 5.7 6.195 6.759

(Z,153,CORR,PER-CENT)

100

46 100

40 32 100

56 43 37 100

37 29 25 34 100

56 44 38 53 35 100

37 30 26 35 24 35 100

40 32 28 37 26 38 26 100

55 44 38 51 35 52 35 38 100

57 45 38 53 35 54 36 38 53 100

57 45 39 54 35 55 36 39 53 55 100

52 41 36 49 33 50 33 36 49 51 52 100

58 45 39 54 35 54 36 38 53 55 55 51 100

49 38 33 46 30 47 31 33 46 47 48 44 48

100

46 36 31 44 29 44 29 31 43 45 45 42 45

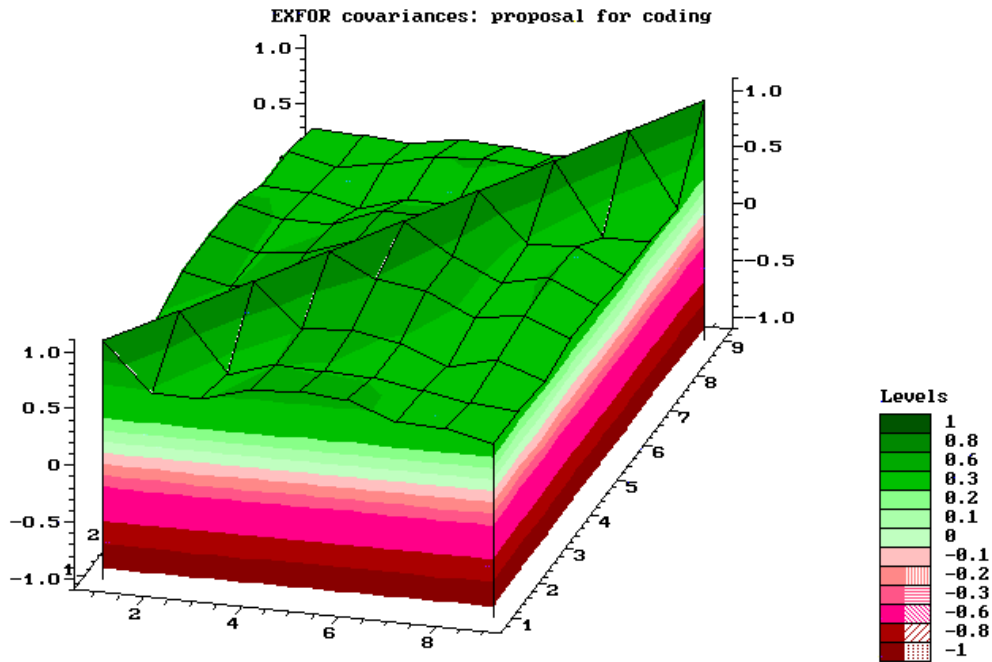
39 100

43 34 29 10 27 41 27 29 40 42 42 39 42

37 35 100

39 31 26 37 24 38 25 26 37 38 39 36 38

34 32 30 100



2) Correlation between different reactions in different subentries (EXFOR: 13134)

123456789.123456789.123456789.123456789.123456789.123456789.123456

COVARIANCE Correlation matrix for fission cross section ratios

(XY,9,Reaction-Ratio,N)

N Reaction Ratio

1 230Th/235U

2 232Th/235U

3 233U/235U

4 234U/235U

5 236U/235U

6 238U/235U

7 237Np/235U

8 239Pu/235U

9 242Pu/235U

(Z,45,CORR,PER-CENT)

100

57 100

58 48 100

69 60 62 100

71 59 64 78 100

68 58 58 73 73 100

53 51 41 53 53 56 100

52 43 50 59 62 59 36 100

42 40 37 44 43 39 37 35 100

There is already Web interface for display of EXFOR covariances coded in this style using Web-ZVView package, where user can input matrix in almost free format, display it with various options and produce output in simple text, html table, ENDF MF33 and draft for EXFOR. What is also important for general users that system produce data for reading in a program including Fortran code together with data. This should attract wide range of users to write applications if they have interest in this field.

Conclusion

The proposal of coding of EXFOR covariance data is trivial, easy for programming, extendable, compatible with existing software and can be easy implemented for all data existing now in EXFOR database.

Output formats from Web-ZVView plotting

See: plotted data out:x4,T,F,e6

```

94242.0000 239.979400      0      0      0      1944633 18  1
      0.0      0.0      0      18      0      1944633 18  2
      0.0      0.0      1      5      171     18944633 18  3
1.00000-05 5.97000+05 7.97000+05 8.98000+05 9.06000+05 1.01000+06944633 18  4
1.09800+06 1.11700+06 1.19300+06 1.37600+06 1.39000+06 1.69300+06944633 18  5
2.14400+06 3.90800+06 5.18200+06 5.70000+06 6.19500+06 6.75900+06944633 18  6
  
```

```

100
46 100
40 32 100
56 43 37 100
37 29 25 34 100
56 44 38 53 35 100
37 30 26 35 24 35 100
40 32 28 37 26 38 26 100
55 44 38 51 35 52 35 38 100
57 45 38 53 35 54 36 38 53 100
57 45 39 54 35 55 36 39 53 55 100
52 41 36 49 33 50 33 36 49 51 52 100
58 45 39 54 35 54 36 38 53 55 55 51 100
49 38 33 46 30 47 31 33 46 47 48 44 48 100
46 36 31 44 29 44 29 31 43 45 45 42 45 39 100
43 34 29 10 27 41 27 29 40 42 42 39 42 37 35 100
39 31 26 37 24 38 25 26 37 38 39 36 38 34 32 30 100
  
```

```

#ZVView-data-copy: 27-Sep-2010 19:09:42
#IRDF-2002:Fe-54(n,n2n)
Z(13x13): Zij = Cor(σXi,σYj)*100
  
```

X (MeV)	13.6	14	14.5	15	15.5	16	16.5	17	17.5	18	19	20	21	J
13.6	100	36.97	41.24	36.67	28.75	18.86	10.54	6.076	5.228	8.462	12	13.2	13.2	I
14	36.97	100	80.2	59.35	39.4	22.1	11.03	6.95	8.029	14.2	17.88	17.68	17.68	2
14.5	41.24	80.2	100	89.93	71.9	46.62	24.3	11.71	8.794	17.78	29.62	34.66	34.66	3
15	36.67	59.35	89.93	100	91.63	70.94	47.08	30.1	21.64	21.52	26.35	29.93	29.93	4
15.5	28.75	39.4	71.9	91.63	100	91.4	74.08	57.53	44.48	28.98	15.99	12.35	12.35	5
16	18.86	22.1	46.62	70.94	91.4	100	93.89	82.62	68.6	39.79	7.189	-5.971	-5.971	6
16.5	10.54	11.03	24.3	47.08	74.08	93.89	100	95.97	85.37	52.41	7.833	-12.77	-12.77	7
17	6.076	6.95	11.71	30.1	57.53	82.62	95.97	100	95.48	67.54	20.24	-4.5	-4.5	8
17.5	5.228	8.029	8.794	21.64	44.48	68.6	85.37	95.48	100	84.81	43.43	17.88	17.88	9
18	8.462	14.2	17.78	21.52	28.98	39.79	52.41	67.54	84.81	100	83.7	65.26	65.26	10
19	12	17.88	29.62	26.35	15.99	7.189	7.833	20.24	43.43	83.7	100	95.87	95.87	11
20	13.2	17.68	34.66	29.93	12.35	-5.971	-12.77	-4.5	17.88	65.26	95.87	100	95.87	12
21	13.2	17.68	34.66	29.93	12.35	-5.971	-12.77	-4.5	17.88	65.26	95.87	95.87	100	13
I	J	2	3	4	5	6	7	8	9	10	11	12	13	

```

#matrix2 for Fortran. ZVView-data-copy: 27-Sep-2010 19:09:42
13 13 169
1.36e+7 1.4e+7 1.45e+7 1.5e+7 1.55e+7 1.6e+7
.....
#//fortran program for reading
integer lx,ly,lz
character*80 filename
character*80 title
real xx(200),yy(200),zz(40000)
in=30
filename='m2.txt'
write (*,*) 'file: ',trim(filename)
open (unit=in, file=filename, status='old')
read (in,*) title
read (in,*) lx,ly,lz
read (in,*) (xx(ii),ii=1,lx)
read (in,*) (yy(ii),ii=1,ly)
read (in,*) (zz(ii),ii=1,lz)
close(unit=in)
call matrix2out(lx,ly,xx,yy,zz)
stop

subroutine matrix2out(lx,ly,xx,yy,zz)
dimension xx(lx),yy(ly),zz(lx,ly)
write (*,*) ' LX=',lx,' LY=',ly,' LZ=',lx*ly
write (*,200) (' i=',ii,' x=',xx(ii),ii=1,lx)
write (*,200) (' j=',jj,' y=',yy(jj),jj=1,ly)
200 format(' ',a3,i3,' ',a3,g13.7)
do jj=1,ly
do ii=1,lx
write (*,*) jj,ii,' x=',xx(ii),' y=',yy(ii),' z=',zz(ii,jj)
enddo
enddo
return
end
  
```

Output for Fortran users: data + code in one file

```

#matrix2 for Fortran. ZVView-data-copy: 27-Sep-2010 19:09:42
13 13 169
1.36e+7 1.4e+7 1.45e+7 1.5e+7 1.55e+7 1.6e+7
.....
#//fortran program for reading
integer lx,ly,lz
character*80 filename
character*80 title
real xx(200),yy(200),zz(40000)
in=30
filename='m2.txt'
write (*,*) 'file: ',trim(filename)
open (unit=in, file=filename, status='old')
read (in,*) title
read (in,*) lx,ly,lz
read (in,*) (xx(ii),ii=1,lx)
read (in,*) (yy(ii),ii=1,ly)
read (in,*) (zz(ii),ii=1,lz)
close(unit=in)
call matrix2out(lx,ly,xx,yy,zz)
stop

subroutine matrix2out(lx,ly,xx,yy,zz)
dimension xx(lx),yy(ly),zz(lx,ly)
write (*,*) ' LX=',lx,' LY=',ly,' LZ=',lx*ly
write (*,200) (' i=',ii,' x=',xx(ii),ii=1,lx)
write (*,200) (' j=',jj,' y=',yy(jj),jj=1,ly)
200 format(' ',a3,i3,' ',a3,g13.7)
do jj=1,ly
do ii=1,lx
write (*,*) jj,ii,' x=',xx(ii),' y=',yy(ii),' z=',zz(ii,jj)
enddo
enddo
return
end
  
```