Title

STOCAL-II: Computer Assisted Learning System for Stochastic Dynamic Analysis of Structures, Part II - User's Manual

Authors

Wung, Chih-Dao
Der Kiureghian, Armen

Publication Date

1989-04-01

REPORT NO.
UCB/SEMM-89/11

STRUCTURAL ENGINEERING,
MECHANICS AND MATERIALS

STOCAL-II: COMPUTER-ASSISTED LEARNING SYSTEM FOR STOCHASTIC DYNAMIC ANALYSIS OF STRUCTURES

PART II -- USER'S MANUAL

By
CHIH-DAO WUNG
and
ARMEN DER KIUREGHIAN

APRIL 1989

DEPARTMENT OF CIVIL ENGINEERING
UNIVERSITY OF CALIFORNIA AT BERKELEY
BERKELEY, CALIFORNIA

STOCAL–II
COMPUTER ASSISTED LEARNING SYSTEM FOR STOCHASTIC DYNAMIC ANALYSIS OF STRUCTURES

PART II
USER'S MANUAL

By

Chih-Dao Wung

and

Armen Der Kiureghian

Report No. UCB/SEMM-89/11
Structural Engineering, Mechanics and Materials
Department of Civil Engineering
University of California, Berkeley

April 1989

TABLE OF CONTENTS

1. INTRODUCTION

STOCAL–II (Computer Assisted Learning of STOchastic methods) is an instructional software developed for the purpose of teaching random vibrations and applied stochastic processes. It is also useful in an engineering office environment for self-learning as well as for solving practical problems involving random vibrations or stochastic processes. STOCAL–II is based on and is an extension of the well known deterministic instructional software CAL, which has facilities for matrix operations and for static and dynamic structural analysis, including static condensation, and eigenvalue solution.

The software works on IBM-PC/XT, AT, or PS2 (or compatible) microcomputers with at least 470 KB free memory. STOCAL–II works both interactively and in batch mode, and has on-line graphics capability. Since IBM's Graphics Development Toolkit is used to develop its graphics facility, proper VDI device drivers for the display monitor and plotter must be installed. The "read.me" file on the diskette containing the STOCAL–II executable files describes the procedure and requirements for installing and using STOCAL–II.

This report contains detailed descriptions of the STOCAL–II commands. The background theory and development of software are described in the companion report†. The CAL commands are not described in this report. The can be found in a separate report by E. L. Wilson‡. Summaries of the command description can also

† "STOCAL–II: Computer-Assisted Learning System for Stochastic Dynamic Analysis of Structures, Part I – Theory and Development," C.-D. Wung and A. Der Kiureghian, Report No. UCB/SEMM-89/10, Department of Civil Engineering, University of California, Berkeley, CA., 1989

‡ "CAL86 – Computer Assisted Learning of Structural Analysis and The CAL/SAP Devel-

be viewed on the monitor by issuing the *HELP* command.

STOCAL–II includes more than 40 commands for stochastic analysis, which are in addition to the commands in CAL. These are categorized in eleven groups as follows:

(1) two dimension graphics,

(2) generation of random numbers and processes,

(3) transformation of samples,

(4) Estimation of samples,

(5) frequency-time domain transformation,

(6) response PSD function of linear systems,

(7) response correlation function of linear systems,

(8) spectral moments,

(9) statistics of stationary Gaussian process,

(10) statistics of nonstationary process, and

(11) Miscellaneous.

A summary of the commands in each group is provided in Section 6. The detailed descriptions of the individual commands are listed in the alphabetical order.

A series of example applications of STOCAL–II commands can be found in Chapter 5 of the companion report. The input batch file for the examples is included on the diskette containing STOCAL–II executable files.

---

opment System," E. L. Wilson, Report No. UCB/SEMM-86/05, University of California, Berkeley, CA., 1986

2. COMMAND SYNTAX

STOCAL–II commands have the following syntax:

COMMANDNAME **Input Matrices** **Output Matrices**+ (**Conditional Matrix**) \

Required Parameters (*Conditional Parameters*) [*Optional Parameters*]

The following font and sign convention is used to define the various parts of the command line:

COMMAND NAME	Slanted
Matrix	Bold
Parameters	Italic
[ ]	Optional
( )	Conditional
+	A newly generated matrix
–	A modified matrix
\	Line continuation

Each parameter is defined in terms of a one or two-character identifier and a list of parameter values in the form:

P=p1,p2,...,pn

where

P	Parameter identifier
p1,p2,...,pn	Parameter values

3. RULES FOR FREE FORMAT INPUT

STOCAL–II uses a free format convention with the following rules:

1. A "C" in column 1 of any input line causes the line to be echoed as a comment on the console.
2. A backslash "\" at the end of an input command line allows the command line to be continued on the following line. A command line can be continued up to a limit of 160 characters.
3. If fewer data are provided than are required, depending on the type of data the remaining items are taken to be either zero or blank.
4. More than one character may be used as a parameter identifier. However, only the last character before the equal sign is treated as the identifier and the other characters are ignored. Therefore, the last character of each identifier must be unique on a given line.
5. When an identifier is not found, default parameter values are used. If no default values exist, previously assigned values of the parameters are used.
6. Real data do not require decimal points. E formats with + or − exponents can be used.
7. Simple arithmetic statements may be used within the input line. The functions that can be used are +, −, *, and /. However, the order of evaluation is sequential, not hierarchical as in the FORTRAN language.
8. Upper or lower case letters may be used equally to define commands, identifiers, or matrices.

4. BASIC COMMANDS

The following is a list of basic commands for matrix operations or data management. Most of these commands are available through CAL.

DELETE **M–** or *D* **M–**

deletes the array **M** in the internal database and releases the storage.

HELP **commandname** or *H* **commandname**

displays a description of the specified command on the monitor.

IF **M1** **M2**

if the absolute value of **M1**(1,1) is less than **M2**(1,1), the *RETURN* command is executed and the *SUBMIT* operation is terminated.

LIST [**DFILE**] or *L* [**DFILE**]

lists the names and sizes of all arrays in the external data file **DFILE**.COR. When **DFILE** is not specified, the arrays in the internal database are listed.

LOAD **M+** *R*=? *C*=?

creates a matrix **M** with *R* rows and *C* columns. The data must be supplied one row per line, and must immediately follow the *LOAD* command. The data is separated by commas or one or more blanks. A line of data may be continued by the use of a backslash at the end of the first line. However, the total length of the line may not exceed 160 characters. If the data for a row is greater than 160 characters, the matrix must be loaded by the use of submatrix operations.

PRINT **M** or *P* **M**

displays the contents of matrix **M** on the console and in the default output file.

QUIT or *Q*

quits the program without saving any data.

READC [**DFILE**] [(**M1**+)]

reads all arrays (or only the array **M1**) from the external data file **DFILE**.COR and puts them (or **M1**) into the internal database. The original names of the arrays are retained. If **DFILE** is not supplied, the default data file name is used.

RETURN

terminates the execution of a submitted batch file (see command *SUBMIT*) and returns to the interactive mode.

SAVE [**DFILE**]

saves all arrays in the internal database to the external data file **DFILE**.COR. The default name is used if **DFILE** is not provided,.

START **DFILE**

initializes the program and its internal database. **DFILE** is assigned as the default name of the database.

STOP or *S*

terminates the use of the program and returns the control to the computer's operating system. All arrays in the internal database are stored in the external data file with the default name.

SYS **doscommand**

allows the user to ISSUE system DOS command **doscommand**. This command can be used, for example, to edit files without leaving STOCAL–II.

SUBMIT **SEP** [**CFILE**] [*N*=*n*]

causes the execution of commands listed in the external file **CFILE** following the separator **SEP** until the command *RETURN* is encountered. The command sequence will be executed *n* times unless terminated by an *IF* operation. The separator name must be in upper case, start in column one, and not start with the letter "C". Default values are **CFILE**=default name, and *N*=1.

5. FUNCTION COMMANDS

The following function operations are available in STOCAL–II:

LOG **M1–**	$\mathbf{M1}(i,j) = ln[\mathbf{M1}(i,j)]$
SQREL **M1–**	$\mathbf{M1}(i,j) = \sqrt{\mathbf{M1}(i,j)}$
INVEL **M1–**	$\mathbf{M1}(i,j) = 1/\mathbf{M1}(i,j)$
EXP **M1–**	$\mathbf{M1}(i,j) = exp[\mathbf{M1}(i,j)]$
COS **M1–**	$\mathbf{M1}(i,j) = cos[\mathbf{M1}(i,j)]$
SIN **M1–**	$\mathbf{M1}(i,j) = sin[\mathbf{M1}(i,j)$
ABS **M1–**	$\mathbf{M1}(i,j) = \lvert\mathbf{M1}(i,j)\rvert$
ACOS **M1–**	$\mathbf{M1}(i,j) = cos^{-1}[\mathbf{M1}(i,j)]$
ASIN **M1–**	$\mathbf{M1}(i,j) = sin^{-1}[\mathbf{M1}(i,j)]$
ATAN **M1–**	$\mathbf{M1}(i,j) = tan^{-1}[\mathbf{M1}(i,j)]$
POW **M1–** *N=n*	$\mathbf{M1}(i,j) = \mathbf{M1}(i,j)^n$
ADD **M1– M2**	$\mathbf{M1}(i,j) = \mathbf{M1}(i,j) + \mathbf{M2}(i,j)$
SUB **M1– M2**	$\mathbf{M1}(i,j) = \mathbf{M1}(i,j) - \mathbf{M2}(i,j)$
MUL **M1– M2**	$\mathbf{M1}(i,j) = \mathbf{M1}(i,j) * \mathbf{M2}(i,j)$
DIV **M1– M2**	$\mathbf{M1}(i,j) = \mathbf{M1}(i,j)/\mathbf{M2}(i,j)$
POW **M1– M2**	$\mathbf{M1}(i,j) = \mathbf{M1}(i,j)^{\mathbf{M2}(i,j)}$
SCALE **M1– M2**	$\mathbf{M1}(i,j) = \mathbf{M1}(i,j) * \mathbf{M2}(1,1)$

In the above, the matrices **M1** and **M2** should have the same number of rows, except in command *SCALE*.

6. STOCAL-II COMMAND SUMMARY

STOCAL-II commands are categorized into the following groups:

(1) Two Dimension Graphics

PLOT draws 2-D curves on the screen and the plotter by supplying the x and y coordinates in a single matrix or two separate matrices. A series of secondary commands are available to draw axes, zoom, etc.

(2) Generation of Samples

GSU generates random numbers between 0 and 1 with uniform distribution.

GSGP generates an ensemble of sample functions for a stationary Gaussian process with a specified PSD function.

GSGPT generates an ensemble of sample functions for a stationary Gaussian process with a specified autocorrelation function.

GEGP generates an ensemble of sample functions for a Gaussian process with an evolutionary PSD function.

TSSF multiplies generated sample functions by a time modulating function.

GPSD discretizes a specified PSD function.

(3) Transformation of Samples

TFSU transforms a uniformly distributed sample to a sample with a specified distribution.

TTSU transforms a sample with a specified distribution to a uniformly distributed sample.

(4) Estimation of Samples

STAT computes the means, standard deviations and skewness coefficients of specified samples of random variables.

NFD constructs the normalized frequency diagram of a given sample.

NCFD constructs the normalized cumulative frequency diagram of a given sample.

ACF computes the ensemble autocorrelation function of a random process from specified sample functions.

TACF computes the temporal autocorrelation function of a random process from a specified sample function.

(5) Fourier Transform

FTP computes the Fourier transform of a piecewise linear function.

IFTP computes the inverse Fourier transform of a piecewise linear function.

FTD computes the Fourier transform for discrete data.

IFTD computes the inverse Fourier transform for discrete data.

(6) Response PSD Functions

SPSD computes the stationary response PSD function.

TPSD computes the evolutionary response PSD function, where the input is specified by a uniformly modulated PSD function.

EPSD computes the evolutionary response PSD function, where the input is specified by an evolutionary PSD function.

(7) Response Correlation Functions

SCF computes the stationary response auto or cross-correlation function.

TCF computes the evolutionary response auto or cross-correlation function, where the input is specified by a uniformly modulated PSD function.

ECF computes the evolutionary response auto or cross-correlation function, where the input is specified by an evolutionary PSD function.

(8) Spectral Moments

SM computes the spectral moments for a specified PSD function.

SRSM computes the spectral moments of a stationary response when the input is specified by a PSD function.

SMSM computes the spectral moments for stationary modal responses when the input is specified by a PSD function.

SMR computes the spectral moments of a stationary response by superposition of modal spectral moments.

RCQC computes the mean of absolute maximum of a response quantity using the CQC response spectrum method.

RSM computes the spectral moments of the response when the input is specified by a mean response spectrum.

(9) Statistics of Stationary Gaussian Process

SSGP computes various statistics of a stationary Gaussian process, including crossing rates, distributions of peaks and the statistics of the envelope process.

LPKD computes the PDF and CDF of the local peaks of a stationary Gaussian process.

EXTD computes the PDF and CDF of the extreme peak of a stationary Gaussian process.

(10) Statistics of Nonstationary Process

TMS computes the variances and cross-correlation coefficients of a uniformly modulated process and its derivatives.

TRMS computes the variances and cross-correlation coefficients of the response and/or its derivatives when the input is specified by a uniformly modulated PSD function.

EMS computes the variances and cross-correlation coefficients of an evolutionary process and its derivatives.

ERMS computes the variances and cross-correlation coefficients of the response and/or its derivatives when the input is specified by an evolutionary PSD function.

NCR computes the mean upcrossing rate of a zero-mean nonstationary Gaussian process above specified thresholds.

NDLP computes the PDF of the local peaks of a zero-mean nonstationary Gaussian process.

NDEP computes the PDF and CDF of the extreme peak of a zero-mean nonstationary Gaussian process.

(11) Miscellaneous

AMP transforms complex numbers expressed by real and imaginary parts into an amplitude and phase angle expression.

MPF computes modal participation factors.

EPF computes modal effective participation factors.

VECTOR constructs a vector containing a sequence of equally spaced ascending numbers.

WRITE writes numerical data onto an external file.

7. DESCRIPTION OF STOCAL–II COMMANDS

The following pages include detailed descriptions of the STOCAL–II commands. Commands are listed in alphabetical order. The description (without the mathematical expressions) are also available on line by use of the *HELP* command.

ACF **MSF** **MCF+** [*DT=dt* *IC=ic*]

computes the ensemble autocorrelation function of a random process based on n sample functions supplied in matrix **MSF**, where

MSF	is an n x nt matrix containing n sample functions each specified at nt equally time points.
MCF+	is an nt x 2 matrix containing the time lags in the first column and the computed autocorrelation values in the second column.
DT=dt	is the time increment of the sample function. (Default: *DT*=1)
IC=ic	is the column index for the reference time. (Default: *IC*=1)

$$\mathbf{MCF}(i,1) = (i - ic) * dt$$

$$\mathbf{MCF}(i,2) = \frac{1}{n}\sum_{k=1}^{n}\mathbf{MSF}(k,i) * \mathbf{MSF}(k,ic)$$

AMP **MGW** **MW+**

transforms complex values with the real and imaginary parts in matrix **MGW** into the amplitude and phase angle expression in matrix **MW**, where

MGW is a three columns matrix containing a sequence of coordinates in the first column and corresponding complex ordinates with the real and imaginary parts in the second and third columns, respectively.

MW+ is a three columns matrix containing a sequence of coordinates in the first column (the same as in matrix **MGW**) and corresponding complex ordinates with the amplitudes and phase angles (between $-\pi$ and π *rad*) in the second and third columns, respectively.

$$MW(i,1) = MGW(i,1)$$

$$MW(i,2) = \sqrt{MGW(i,2)^2 + MGW(i,3)^2}$$

$$\begin{aligned} MW(i,3) &= \pi - \theta_i \qquad MGW(i,2) < 0 \text{ and } MGW(i,3) \geq 0 \\ &= -\pi - \theta_i \qquad MGW(i,2) < 0 \text{ and } MGW(i,3) < 0 \\ &= \theta_i \qquad \text{elsewhere} \end{aligned}$$

where $\theta_i = sin^{-1}(\frac{MGW(i,3)}{MW(i,2)})$

ECF **VW VD MZ1 MZ2 MWT MCF+ (MPI)** *I=type* *P=p1,p2,...* \
T1=t1b,t1e [*T2=t2b,t2e* *N=nt* *M=m1,m2* *IC=i1,i2* *L=l*]

computes the auto or cross-correlation function of response quantities Z1(*t1*) and Z2(*t2*) or their time derivatives for an input excitation specified by an evolutionary PSD function, i.e. a PSD of type *I* modulated by the time-frequency function **MWT**. Autocorrelation is computed when Z1 and Z2 denote the same response quantity; otherwise, cross-correlation is computed, where

VW(ω_i)	is an *n* vector containing the modal frequencies in *rad/sec*.
VD(ζ_i)	is an *n* vector containing the modal damping ratios.
MZ1 a_{1i}	is an *n* rows matrix, where the *i1*-th column contains the modal effective participation factors for the response Z1.
MZ2 a_{2j}	is an *n* rows matrix, where the *i2*-th column contains the modal effective participation factors for the response Z2.
MWT $A(\omega,\tau)$	is a *k* x *p* matrix specifying a time-frequency modulating function, where the first column contains a sequence of ascending time coordinates, the first row contains a sequence of non-negative, ascending frequency coordinates, and the remainder of the matrix contains the ordinates of the modulating function for the specified time and frequency coordinates. Between the specified points the function is assumed to behave linearly. The first entry of the matrix must be −1.1.
MCF+ $\phi_{z_1,z_2}^{(m_1,m_2)}(t_1,t_2)$	is an *nt* x 3 matrix, where the first and second columns contain the time coordinates *t1* and *t2*, respectively, and the third column contains the corresponding auto or cross-correlation values.

MPI $\Phi_{ff}(\omega)$ — is needed only for *I*=3 to specify the input PSD function. This is a two or three columns matrix, where the first column contains a sequence of non-negative, ascending frequency coordinates while the second and third columns contain the corresponding real and imaginary parts of the PSD function. If the imaginary part is zero, only two columns are needed. The real part of **MPI** is assumed to be an even function while the imaginary part is assumed to be an odd function. The PSD is assumed to be zero outside the specified frequency interval.

I=type — *I*=1: White Noise

I=2: Banded Linear Noise

I=3: Arbitrary piecewise linear PSD specified by **MPI**.

P=p1,p2,... — are PSD parameters as follows:

I=1: *p1* is the amplitude. (Default: *p1*=1)

I=2: *p1* is the amplitude at the lower frequency bound, *p2* is the lower frequency bound, *p3* is the upper frequency bound, and *p4* is the amplitude at the upper frequency bound. The function between the bounds is linear and outside the bounds is zero.

I=3: *p1* is a scaler multiplier of **MPI**. (Default: *p1*=1)

T1=t1b,t1e — are the lower and upper bounds for time coordinates *t*1 associated with the first response quantity.

T2=t2b,t2e — are the lower and upper bounds for time coordinates *t*2 associated with the second response quantity. (Default: *T2=T1*)

N=nt — is the number of time points between *t1b* and *t1e* and between *t2b* and *t2e*. (Default: *nt*=2)

M=m1,m2 are the derivative orders for the two response quantities, respectively. (Default: *m1=m2*=0)

IC=i1,i2 are the column indices of matrices **MZ1** and **MZ2**, which contain the modal effective participation factors for response Z1 and Z2, respectively. (Default: *i1=1=i2*=1)

L=l is the number of modes considered. (Default: *l*=n)

$$\phi_{z_1,z_2}^{(m_1,m_2)}(t_1,t_2) = \sum_{i=1}^{l}\sum_{j=1}^{l} a_{1i} \; a_{2j} \; \phi_{ij}^{(m_1,m_2)}(t_1,t_2)$$

where

$$\phi_{ij}^{(m_1,m_2)}(t_1,t_2) = \int_{-\infty}^{\infty} \frac{\partial^{m_1}[\int_0^{t_1} A(\omega,\tau)h_i(t_1-\tau)e^{i\omega\tau}d\tau]}{\partial t_1^{m_1}} \frac{\partial^{m_2}[\int_0^{t_2} A(\omega,\tau)h_j(t_2-\tau)e^{-i\omega\tau}d\tau]}{\partial t_2^{m_2}} \Phi_{ff}(\omega)d\omega$$

where $h_i(t)$ is the unit impulse response function of mode i.

EMS **MWT VR+ (MPI)** *I=type P=p1,p2,... T=tb,te [N=nt]*

computes the variances and cross-correlation coefficients of an evolutionary process and its derivatives. The evolutionary process is described by an evolutionary PSD function, i.e. a PSD of type I modulated by the time-frequency function **MWT**, where

MWT $A(\omega, t)$ — is a k x p matrix specifying a time-frequency modulating function, where the first column contains a sequence of ascending time coordinates, the first row contains a sequence of non-negative, ascending frequency coordinates, and the remainder of the matrix contains the ordinates of the modulating function for the specified time and frequency coordinates. Between the specified points the function is assumed to behave linearly. The first entry of the matrix must be -1.1.

VR+ — is an nt x 7 matrix, where the first column contains the time coordinates and the remaining six columns store the variances of the derivatives of order 0, 1 and 2, and the cross-correlation coefficients of the derivatives of orders 0 and 1, 0 and 2, and 1 and 2, respectively.

MPI $\Phi(\omega)$ — is needed only for I=3 to specify the input PSD function. This is a two or three columns matrix, where the first column contains a sequence of non-negative, ascending frequency coordinates while the second and third columns contain the corresponding real and imaginary parts of the PSD function. If the imaginary part is zero, only two columns are needed. The real part of **MPI** is assumed to be an even function while the imaginary part is assumed to be an odd function. The PSD is assumed to be zero outside the specified frequency interval.

I=type — *I*=1: White Noise

I=2: Banded Linear Noise

I=3: Arbitrary piecewise linear PSD specified by **MPI**.

P=p1,p2,... — are PSD parameters as follows:

I=1: *p1* is the amplitude. (Default: *p1*=1)

I=2: *p1* is the amplitude at the lower frequency bound, *p2* is the lower frequency bound, *p3* is the upper frequency bound, and *p4* is the amplitude at the upper frequency bound. The function between the bounds is linear and outside the bounds is zero.

I=3: *p1* is a scaler multiplier of **MPI**. (Default: *p1*=1)

T=tb,te — are the lower and upper bounds of equally spaced time coordinates.

N=nt — is the number of time coordinates. (Default: *nt*=2)

$$\phi_{xx}^{(m1,m2)}(t,t) = \int_{-\infty}^{\infty} \frac{\partial^{m1} A(\omega,t)e^{i\omega t}}{\partial t^{m1}} \frac{\partial^{m2} A(\omega,t)e^{-i\omega t}}{\partial t^{m2}} \Phi(\omega)d\omega$$

EPF **MPHI VP MQ MA+**

computes the modal effective participation factors of an MDOF system for a given response transfer matrix, where

MPHI is an n x m matrix containing the m mode shapes of an n-DOF system.

VP is an m vector containing the modal participation factors.

MQ is an l x n response transfer matrix, where n is the number of degrees of freedom and l is the number of response quantities of interest. This matrix relates the degrees of freedom to the response quantities of interest.

MA+ is an m x l matrix containing the resulting modal effective participation factors for the l response quantities and for each of the m modes.

$$\mathbf{MA}(i,j) = \sum_{k=1}^{n} \mathbf{MQ}(j,k) * \mathbf{MPHI}(k,i) * \mathbf{VP}(i)$$

EPSD **VW VD MZ1 MZ2 MWT MPO+ (MPI)** *I=type P=p1,p2,...* [N=nw\
W=wb,we] *T1=t11,t12,...* [*T2=t21,t22,... M=m1,m2 C=i1,i2* L=l]

computes the evolutionary auto or cross-PSD function of response quantities Z1($t1$) and Z2($t2$) or their time derivatives for an input excitation specified by an evolutionary PSD function, i.e. a PSD of type *I* modulated by the time-frequency function **MWT**. Auto-PSD is computed when Z1 and Z2 denote the same response quantity; otherwise, cross-PSD is computed, where

VW(ω_i) — is an n vector containing the modal frequencies in rad/sec.

VD(ς_i) — is an n vector containing the modal damping ratios.

MZ1 a_{1i} — is an n rows matrix, where the *i1*-th column contains the modal effective participation factors for the response Z1.

MZ2 a_{2j} — is an n rows matrix, where the *i2*-th column contains the modal effective participation factors for the response Z2.

MWT $A(\omega,\tau)$ — is a k x p matrix specifying a time-frequency modulating function, where the first column contains a sequence of ascending time coordinates, the first row contains a sequence of non-negative, ascending frequency coordinates, and the remainder of the matrix contains the ordinates of the modulating function for the specified time and frequency coordinates. Between the specified points the function is assumed to behave linearly. The first entry of the matrix must be -1.1.

MPO+ $\Phi_{z_1,z_2}^{(m_1,m_2)}(\omega,t_1,t_2)$ — is an nw rows matrix, where the first column of the matrix contains a sequence of nw equally spaced frequency coordinates beginning at *wb* and ending at *we*, while the 2nd, 4th,... columns contain the real parts and the 3rd, 5th,... columns contain the imaginary parts of the resulting PSD functions at the specified times.

MPI $\Phi_{ff}(\omega)$	is needed only for *I*=3 to specify the input PSD function. This is a two or three columns matrix, where the first column contains a sequence of non-negative, ascending frequency coordinates, while the second and third columns contain the corresponding real and imaginary parts of the PSD function. If the imaginary part is zero, only two columns are needed. The real part of **MPI** is assumed to be an even function while the imaginary part is assumed to be an odd function. The PSD is assumed to be zero outside the specified frequency interval.
I=type	*I*=1: White Noise *I*=2: Banded Linear Noise *I*=3: Arbitrary piecewise linear PSD specified by **MPI**. *I*=4: Filtered White Noise
P=p1,p2,...	are PSD parameters as follows: *I*=1: *p1* is the amplitude. (Default: *p1*=1) *I*=2: *p1* is the amplitude at the lower frequency bound, *p2* is the lower frequency bound, *p3* is the upper frequency bound, and *p4* is the amplitude at the upper frequency bound. The function between the bounds is linear and outside the bounds is zero. *I*=3: *p1* is a scaler multiplier of **MPI**. (Default: *p1*=1) *I*=4: *p1* (Φ_o) is the amplitude at zero frequency, and *p2* (ω_g) and *p3* (ς_g) are the natural frequency and damping ratio of the filter, respectively.
W=wb,we	are the lower and upper frequency limits of the output data in *rad/sec*. (Default: *wb*=0, *we*=100)

N=nw	is the number of points to be generated for the output data. (Default: nw = 101).
T1=t11,t12,...	are the time points of interest for the first response quantity.
T2=t21,t22,...	are the time points of interest for the second response quantity. (Default: *T2=T1*)
M=m1,m2	are the derivative orders for the two response quantities, respectively. (Default: *m1=2=*0)
IC=i1,i2	are the column indices of matrices **MZ1** and **MZ2**, which contain the modal effective participation factors for response Z1 and Z2, respectively. (Default: *i1=i2=*1)
L=l	is the number of modes considered. (Default: *l*=n)

$$\Phi_{z_1,z_2}^{(m_1,m_2)}(\omega,t_1,t_2)=\sum_{i=1}^{l}\sum_{j=1}^{l} a_{1i}\ a_{2j}\ \Phi_{ij}^{(m_1,m_2)}(\omega,t_1,t_2)$$

where

$$\Phi_{ij}^{(m_1,m_2)}(\omega,t_1,t_2)=\frac{\partial^{m_1}[\int_0^{t_1} A(\omega,\tau)h_j(t_1-\tau)e^{\mathrm{i}\omega\tau}d\tau]}{\partial t_1^{m_1}}$$
$$\frac{\partial^{m_2}[\int_0^{t_2} A(\omega,\tau)h_k(t_2-\tau)e^{-\mathrm{i}\omega\tau}d\tau]}{\partial t_2^{m_2}}\ \Phi_{ff}(\omega)e^{\mathrm{i}\omega(t_2-t_1)}$$

where $h_i(t)$ is the unit impulse response function of mode i.

ERMS **VW VD MZ MWT VR+ (MPI)** *I=type* *P=p1,p2,...* *T=tb,te* \
[*N=nt* *IC=ic* *L=l*]

computes the variances and cross-correlation coefficients of the response and its first and second derivatives for an input excitation specified by an evolutionary PSD function, i.e. a PSD function of type *I* modulated by the time-frequency function **MWT**, where

VW(ω_i) is an n vector containing the modal frequencies in *rad/sec*.

VD(ς_i) is an n vector containing the modal damping ratios.

MZ a_i is an n rows matrix where the *ic*-th column contains the modal effective participation factors for the response quantity of interest.

MWT $A(\omega,\tau)$ is a k x p matrix specifying a time-frequency modulating function, where the first column contains a sequence of ascending time coordinates, the first row contains a sequence of non-negative, ascending frequency coordinates, and the remainder of the matrix contains the ordinates of the modulating function for the specified time and frequency coordinates. Between the specified points the function is assumed to behave linearly. The first entry of the matrix must be -1.1.

VR+ is an *nt* x 7 matrix, where the first column contains the time coordinates and the remaining six columns store the variances of the derivatives of order 0, 1 and 2, and the cross-correlation coefficients of the derivatives of orders 0 and 1, 0 and 2, and 1 and 2, respectively.

MPI $\Phi_{ff}(\omega)$ — is needed only for *I*=3 to specify the input PSD function. This is a two or three columns matrix, where the first column contains a sequence of non-negative, ascending frequency coordinates while the second and third columns contain the corresponding real and imaginary parts of the PSD function. If the imaginary part is zero, only two columns are needed. The real part of **MPI** is assumed to be an even function while the imaginary part is assumed to be an odd function. The PSD is assumed to be zero outside the specified frequency interval.

I=type — *I*=1: White Noise

I=2: Banded Linear Noise

I=3: Arbitrary piecewise linear PSD specified by **MPI**.

P=p1,p2,... — are PSD parameters as follows:

I=1: ***p1*** is the amplitude. (Default: ***p1***=1)

I=2: ***p1*** is the amplitude at the lower frequency bound, ***p2*** is the lower frequency bound, ***p3*** is the upper frequency bound, and ***p4*** is the amplitude at the upper frequency bound. The function between the bounds is linear and outside the bounds is zero.

I=3: ***p1*** is a scaler multiplier of **MPI**. (Default: ***p1***=1)

T=tb,te — are the lower and upper bounds of equally spaced time coordinates.

N=nt — is the number of time coordinates. (Default: ***nt***=2)

IC=ic — is the column index of matrix **MZ**, which contains the modal effective participation factors for the response quantity of interest. (Default: ***ic=1***)

L=l — is the number of modes considered. (Default: *l*=n)

$$\phi_{x_1,x_2}^{(m_1,m_2)}(t,t) = \sum_{i=1}^{l}\sum_{j=1}^{l} a_{1i}\ a_{2j}\ \phi_{ij}^{(m_1,m_2)}(t,t)$$

where

$$\phi_{ij}^{(m_1,m_2)}(t,t) = \int_{-\infty}^{\infty} \frac{\partial^{m_1}[\int_0^t A(\omega,\tau)h_i(t-\tau)e^{i\omega\tau}d\tau]}{\partial t^{m_1}} \frac{\partial^{m_2}[\int_0^t A(\omega,\tau)h_j(t-\tau)e^{-i\omega\tau}d\tau]}{\partial t^{m_2}}\ \Phi_{ff}(\omega)d\omega$$

where $h_i(t)$ is the unit impulse response function of mode i.

EXTD **VSM** **MD+** *T=τ* [*R=r1,r2* *N=n* *MU=*μ_X]

computes the PDF and CDF of the extreme peak of a stationary Gaussian $X(t)$ process with mean μ_X over a duration τ. When *MU* is not specified, $\mu_X = 0$ is assumed and the extreme peak is defined as $max|X(t)|$. When *MU* is specified, the extreme peak is defined as $max\ X(t)$, where

VSM λ_m — is a 4 vector containing the response spectral moments of order 0, 1, and 2 in the first three addresses.

MD+ — is an n x 3 matrix, where the first column contains the threshold levels, and the second and third columns contain the corresponding ordinates of the PDF and CDF, respectively.

T=τ — is the duration of the process.

R=r1,r2 — are the lower and upper threshold bounds expressed in terms of units of the root-mean-square, i.e., square root of the spectral moment of order 0. (Default: *r1*=0 and *r2*=5)

N=n — is the number of threshold levels to be considered. (Default: *N*=101)

MU=μ_X — is the mean value of the process.

For the case with a specified mean, the extreme peak is defined $max\ X(t)$ and the CDF † is

$$F_{X_\tau}(x) = (1 - e^{-\frac{r^2}{2}})\ exp(-\nu_e\tau\ \frac{1 - e^{-\sqrt{\frac{\pi}{2}}\delta_e r}}{e^{\frac{r^2}{2}} - 1}) \qquad r2 \geq r \geq r1$$

where r $= \frac{x-\mu_X}{\sigma_X}$, $\delta_e = (2\delta)^{1.2}$, $\delta = \sqrt{1 - \frac{\lambda_1^2}{\lambda_0\lambda_2}}$ and $\nu_e = \frac{1}{2\pi}\sqrt{\frac{\lambda_2}{\lambda_0}}$, in which λ_o, λ_1, and λ_2 are the 0th, 1st and 2nd spectral moments. When the extreme peak is defined by

† "On the Distribution of First-Passage Time for Normal Stationary Processes," E. H. Vanmarcke, *Journal of Applied Mechanics*, ASME, 42, pp. 215-220, 1975.

$max|X(t)|$, for a zero mean process, the CDF is defined by the above with $\delta_e = (\delta)^{1.2}$ and $\nu_e = \frac{1}{\pi}\sqrt{\frac{\lambda_2}{\lambda_0}}$.

FTD **MT MW+** [*W=wb,we N=nw I=i P=p*]

computes the Fourier transform of a time function stored in matrix **MT** by use of the discrete Fourier transform. When *nw* is not specified, the fast Fourier transform algorithm is used, where

MT $f(t)$ — is a two or three columns matrix, where the first column contains a sequence of equally spaced time coordinates and the second and third columns contain the corresponding real and imaginary parts of the time function, respectively. The function is assumed to be zero outside the specified time range. When *I*=1 is specified, the function is assumed to be even in its real part and odd in its imaginary part. Only values of the function for positive time need to be specified in that case. When **MT** has only two columns, the imaginary part is assumed to be zero.

MW+ $\bar{f}(\omega)$ — is the resulting two or three columns matrix, where the first column contains a sequence of equally spaced frequency coordinates beginning at *wb* and ending at *we*, and the second and third columns contain the real and imaginary parts of the resulting frequency function, respectively. If the imaginary part is zero, only two columns are given.

W=wb,we — are the lower and upper bounds of the equally spaced frequency coordinates in *rad/sec*. (Default: *wb*=0 and *we*= $\frac{\pi}{\Delta t}$, where Δt is the interval between any two consecutive time coordinates of function **MT**.)

N=nw — is the number of points to be generated for the output data. When *nw* is not specified, the generalized fast Fourier transform algorithm is used, which internally determines approximate values for *nw* and *we*.

I=i	*I*=0:	A regular function, **MT**, is specified.
	I=1:	The function **MT** is even in its real part and odd in its imaginary part. Only values of the function for positive time need to be specified in this case.
	(Default: *I*=0)	
P=p	is a scaler multiplier of function **MT**. (Default: *p=1*)	

When the number of output data nw is specified,

$$\bar{f}(\omega_j) = \frac{\Delta t}{2\pi} \sum_{k=0}^{nt-1} f(t_k) e^{-i\omega_j t_k} \qquad 0 \le j \le nw - 1$$

where k is the row number of **MT** and $\omega_o = wb$ and $\omega_{nw-1} = we$.

When nw is not specified, the fast Fourier transform algorithm is used. In that case, nw=INT($nt/2$), $\Delta\omega = \frac{2\pi}{nt\Delta t}$, and

$$\bar{f}(\omega_o + j\Delta\omega) = \frac{\Delta t}{2\pi} \sum_{k=0}^{nt-1} f(t_o + k\Delta t) e^{-i(\omega_o + j\Delta\omega)(t_o + k\Delta t)} \qquad 0 \le j \le nw - 1$$

FTP **MT** **MW+** [*W=wb,we N=nw I=i P=p*]

computes the Fourier transform of a piecewise linear function **MT**, where

MT $f(t)$ — is a two or three columns matrix, where the first column contains the ascending time coordinates and the second and third columns contain the corresponding real and imaginary parts of the time function, respectively. The function is assumed to be zero outside the specified time range. When *I*=1 is specified, the function is assumed to be even in its real part and odd in its imaginary part. Only values of the function for positive time need to be specified in that case. When **MT** has only two columns, the imaginary part is assumed to be zero.

MW+ $\bar{f}(\omega)$ — is the resulting two or three columns matrix, where the first column contains a sequence of equally spaced frequency coordinates beginning at *wb* and ending at *we*, and the second and third columns contain the real and imaginary parts of the resulting frequency function, respectively. If the imaginary part is zero, only two columns are given.

W=wb,we — are the lower and upper bounds for the equally spaced frequency coordinates in *rad/sec*. (Default: *wb*=*we*=0)

N=nw — is the number of points to be generated for the output data. (Default: *nw*=1)

I=0: A regular function, **MT**, is specified.

I=1: The function **MT** is even in its real part and odd in its imaginary part. Only values of the function for positive time need to be specified in this case.

(Default: *I*=0)

P=p — is a scaler multiplier of function **MT**. (Default: *p=1*)

$$\bar{f}(\omega) = \frac{p}{2\pi} \int_{-\infty}^{\infty} f(t) e^{-\mathrm{i}\omega t} dt \qquad w_b \leq \omega \leq w_e$$

GEGP **MWT** **MX+** (*MPI*) *I=type* *P=p1,p2,...* *N=n,nt* *T=tb,te* [*M=m* *RS=rs*]

generates samples of a nonstationary Gaussian process with a specified evolutionary PSD function, i.e., a PSD of type *I* modulated by the time-frequency function **MWT**, where

MWT $A(\omega, t)$ — is a *k* x *p* matrix specifying a time-frequency modulating function, where the first column contains a sequence of ascending time coordinates, the first row contains a sequence of non-negative, ascending frequency coordinates, and the remainder of the matrix contains the ordinates of the modulating function for the specified time and frequency coordinates. Between the specified points the function is assumed to behave linearly. The first entry of the matrix must be -1.1.

MX+ — is an *n* x *nt* matrix containing *n* sample functions, each specified at *nt* equally spaced time points.

MPI $\Phi(\omega)$ — is needed only for *I*=3 to specify the input PSD function. This is a two or three columns matrix, where the first column contains a sequence of non-negative, ascending frequency coordinates while the second and third columns contain the corresponding real and imaginary parts of the PSD function. If the imaginary part is zero, only two columns are needed. The real part of **MPI** is assumed to be an even function while the imaginary part is assumed to be an odd function. The PSD is assumed to be zero outside the specified frequency interval.

I=type

- I=1: White Noise
- I=2: Banded Linear Noise
- I=3: Arbitrary piecewise linear PSD specified by **MPI**.
- I=4: Filtered White Noise

P=p1,p2,... are PSD parameters as follows:

- I=1: ***p1*** is the amplitude. (Default: ***p1***=1)
- I=2: ***p1*** is the amplitude at the lower frequency bound, ***p2*** is the lower frequency bound, ***p3*** is the upper frequency bound, and ***p4*** is the amplitude at the upper frequency bound. The function between the bounds is linear and outside the bounds is zero.
- I=3: ***p1*** is a scaler multiplier of **MPI**. (Default: ***p1***=1)
- I=4: ***p1*** (Φ_o) is the amplitude at zero frequency, and ***p2*** (ω_g) and ***p3*** (ς_g) are the natural frequency and damping ratio of the filter, respectively.

N=n,nt are the number of sample functions and the number of points specifying each function, respectively.

T=tb,te are the beginning and ending times of the sample functions.

M=m is the number of points to be used in the frequency domain to discretize the PSD function. (Default: ***m***=100)

RS=rs is a random seed between 0.0 and 1.0 used to generate a sequence of random numbers. (Default: ***rs***=0.5)

$$\mathbf{MX}(k,j) = 2\sum_{i=1}^{m} A(\omega_i,t)\sqrt{\Phi(\omega_i)\Delta\omega}\ \cos(\omega_i t + u_{ik}) \qquad t_b \le t \le t_e$$

where u_{ik} are random variables with uniform distribution between $[0,\ 2\pi]$.

GPSD **MP+** **(MPI)** *I=type* *P=p1,p2,...* [*W=wb,we* *N=nw*]

creates matrix **MP** to store discretized data for a specified PSD function, where

MP+ is an *nw* x 2 matrix, where the first column contains a sequence of equally spaced frequency coordinates beginning at *wb* and ending at *we*, while the second column contains the ordinates of the specified PSD function.

MPI $\Phi(\omega)$ is needed only for *I*=3 to specify the input PSD function. This is a two or three columns matrix, where the first column contains a sequence of non-negative, ascending frequency coordinates, while the second and third columns contain the corresponding real and imaginary parts of the PSD function. If the imaginary part is zero, only two columns are needed. The real part of **MPI** is assumed to be an even function while the imaginary part is assumed to be an odd function. The PSD is assumed to be zero outside the specified frequency interval.

I=type

I=1: White Noise

I=2: Banded Linear Noise

I=3: Arbitrary piecewise linear PSD specified by **MPI**.

I=4: Filtered White Noise

P=p1,p2,... are PSD parameters as follows:

I=1: *p1* is the amplitude. (Default: *p1*=1)

I=2: *p1* is the amplitude at the lower frequency bound, *p2* is the lower frequency bound, *p3* is the upper frequency bound, and *p4* is the amplitude at the upper frequency bound. The function between the bounds is linear and outside the bounds is zero.

I=3: ***p1*** is a scaler multiplier of **MPI**. (Default: ***p1***=1)

I=4: ***p1*** (Φ_o) is the amplitude at zero frequency, and ***p2*** (ω_g) and ***p3*** (ζ_g) are the natural frequency and damping ratio of the filter, respectively.

W=wb,we are the lower and upper bounds of the equally spaced frequency coordinates in *rad/sec*. (Default: ***wb***=0, ***we***=100)

N=nw is the number of points to be generated for the output data. (Default: ***nw***=101).

GSGP **MX+** (**MPI**) *I=type P=p1,p2,...* *N=n,nt T=tb,te* [*M=m RS=rs*]

generates sample functions of a zero-mean stationary Gaussian process with a PSD function of type *I*, where

MX+ is an *n* x *nt* matrix containing *n* sample functions, each specified at *nt* equally spaced time points.

MPI $\Phi(\omega)$ is needed only for *I*=3 to specify the input PSD function. This is a two or three columns matrix, where the first column contains a sequence of non-negative, ascending frequency coordinates while the second and third columns contain the corresponding real and imaginary parts of the PSD function. If the imaginary part is zero, only two columns are needed. The real part of **MPI** is assumed to be an even function while the imaginary part is assumed to be an odd function. The PSD is assumed to be zero outside the specified frequency interval.

I=type
- *I*=1: White Noise
- *I*=2: Banded Linear Noise
- *I*=3: Arbitrary piecewise linear PSD specified by **MPI**.
- *I*=4: Filtered White Noise

P=p1,p2,... are PSD parameters as follows:

- *I*=1: *p1* is the amplitude. (Default: *p1*=1)
- *I*=2: *p1* is the amplitude at the lower frequency bound, *p2* is the lower frequency bound, *p3* is the upper frequency bound, and *p4* is the amplitude at the upper frequency bound. The function between the bounds is linear and outside the bounds is zero.

I=3: ***p1*** is a scaler multiplier of **MPI**. (Default: ***p1***=1)

I=4: ***p1*** (Φ_o) is the amplitude at zero frequency, and ***p2*** (ω_g) and ***p3*** (ζ_g) are the natural frequency and damping ratio of the filter, respectively.

N=n,nt are the number of sample functions and the number of points specifying each function, respectively.

T=tb,te are the beginning and ending times of the sample functions.

M=m is the number of points to be used in the frequency domain to discretize the PSD function. (Default: ***m***=100)

RS=rs is a random seed between 0.0 and 1.0 used to generate a sequence of random numbers. (Default: ***rs***=0.5)

$$\mathbf{MX}(k,j) = 2\sum_{i=1}^{m} \sqrt{\Phi(\omega_i)\Delta\omega}\ \ cos(\omega_i t + u_{ik}), \qquad t_b \le t \le t_e$$

where ***uik*** are random variables with uniform distribution between [0, 2π].

GSGPT **MCF MX+** ***N=n,nt T=tb,te*** *[RS=rs]*

generates sample functions of a zero-mean stationary Gaussian process specified by the autocorrelation function **MCF**, where

MCF is a two columns autocorrelation function matrix, where the first column contains the non-negative, ascending time-lag coordinates, while the second column contains the corresponding autocorrelation values. The function is assumed to be an even function of the time lag. Values outside the specified range are assumed to be zero. **MCF** must be a positive definite function.

MX+ is an *n* x *nt* matrix containing the *n* generated sample functions, each specified at *nt* equally spaced time points.

N=n,nt are the number of sample functions and the number of points specifying each function, respectively.

T=tb,te are the beginning and ending times of the sample functions.

RS=rs is a random seed between 0.0 and 1.0 used to generate a sequence of random numbers. (Default: *rs*=0.5)

$$\mathbf{MX} = \mathbf{L}\ \mathbf{u} \quad \text{and} \quad \mathbf{MCF} = \mathbf{L}\ \mathbf{L}^T \quad \text{(Cholesky decomposition)}$$

where **u** is generated as a standard Gaussian vector.

This command is more appropriate for generating a sequence of identically distribution Gaussian variables, rather than long sample functions.

GSU **MSU+** *N=n,ns* [*RS=rs*]

generates *ns* sets of *n* independent random numbers sampled uniformly between 0 and 1, where

MSU+	is an *n* x *ns* matrix containing the random numbers. (This is used to generate other types of random numbers.)
N=n,ns	are the sample size and the number of samples.
RS=rs	is a random seed between 0.0 and 1.0 used to generate a sequence of random numbers. (Default: *rs*=0.5)

The random number generator uses the following algorithm (Schrage 1979).

```
data k,j,m,rm/5701,3612,566927,566927.0/
ix      = int(x(i)*rm)
irand   = mod(j*ix+k,m)
x(i+1)  = (real(irand)+0.5)/rm
```

IFTD **MW** **MT+** [*T=tb,te* *N=nt* *I=i* *P=p*]

computes the inverse Fourier transform of a frequency function **MW** by use of the discrete Fourier transform. When *nt* is not specified, the fast Fourier transform algorithm is used, where

MW $\bar{f}(\omega)$ — is a two or three columns matrix, where the first column contains a sequence of equally spaced frequency coordinates and the second and third columns contain the corresponding real and imaginary parts of the frequency function, respectively. The function is assumed to be zero outside the specified frequency range. When *I*=1 is specified, the function is assumed to be even in its real part and odd in its imaginary part. Only values of the function for positive frequency need to be specified in that case. When **MW** has only two columns, the imaginary part is assumed to be zero.

MT+ $f(t)$ — is the resulting two or three columns matrix, where the first column contains a sequence of equally spaced time coordinates beginning at *tb* and ending at *te*, and the second and third columns contain the real and imaginary parts of the resulting time function, respectively. If the imaginary part is zero, only two columns are given.

T=tb,te — are the lower and upper bounds of the equally spaced time coordinates. (Default: *tb*=0 and *te*= $\frac{\pi}{\Delta\omega}$, where $\Delta\omega$ is the interval between any two consecutive frequency coordinates of function **MW**.)

N=nt — is the number of points to be generated for the output data. When *nt* is not specified, the generalized fast Fourier transform algorithm is used, which internally determines approximate values for *nt* and *te*.

I=*i* — *I*=0: A regular function, **MW**, is specified.

I=1: The function **MW** is even in its real part and odd in its imaginary part. Only values of the function for positive frequency need to be specified in this case.

(Default: *I*=0)

P=*p* — is a scaler multiplier of function **MW**. (Default: *p*=1)

When the number of output data nt is specified,

$$f(t_j) = \Delta\omega \sum_{k=0}^{nw-1} \bar{f}(\omega_k) e^{-it_j\omega_k} \qquad 0 \leq j \leq nt-1$$

where k is the row number of **MW** and $t_o = tb$ and $t_{nt-1} = te$.

When nt is not specified, the fast Fourier transform algorithm is used. In that case, nt=INT($nw/2$), $\Delta t = \frac{2\pi}{nw\Delta\omega}$, and

$$f(t_o + j\Delta t) = \Delta\omega \sum_{k=0}^{nw-1} \bar{f}(\omega_o + k\Delta\omega) e^{-i(t_o+j\Delta t)(\omega_o+k\Delta\omega)} \qquad 0 \leq j \leq nt-1$$

IFTP **MW** **MT+** [T=*tb*,*te* N=*nt* I=*i* P=*p*]

computes the inverse Fourier transform of a piecewise linear function **MW**, where

MW $\bar{f}(\omega)$	is a two or three columns matrix, where the first column contains the ascending frequency coordinates and the second and third columns contain the corresponding real and imaginary parts of the frequency function, respectively. The function is assumed to be zero outside the specified frequency range. When I=1 is specified, the function is assumed to be even in its real part and odd in its imaginary part. Only values of the function for positive frequency need to be specified in that case. When **MW** has only two columns, the imaginary part is assumed to be zero.
MT+ $f(t)$	is the resulting two or three columns matrix, where the first column contains a sequence of equally spaced time coordinates beginning at *tb* and ending at *te*, and the second and third columns contain the real and imaginary parts of the resulting time function, respectively. If the imaginary part is zero, only two columns are given.
T=*tb*,*te*	the lower and upper bounds of the equally spaced time coordinates. (Default: *tb*=*te*=0)
N=*nt*	is the number of points to be generated for the output data. (Default: *nt*=1)
I=*i*	I=0: A regular function, **MW**, is specified. I=1: The function **MW** is even in its real part and odd in its imaginary part. Only values of the function for positive frequency need to be specified in this case. (Default: I=0)

P=p	is a scaler multiplier of function **MW**. (Default: *p*=1)

$$f(t) = p \int_{-\infty}^{\infty} \bar{f}(\omega) e^{i\omega t} d\omega \qquad t_b \leq t \leq t_e$$

LPKD **VSM** **MD+** [*R=r1,r2* *N=n*]

computes the PDF and CDF of the local peaks of a zero-mean stationary Gaussian process, where

VSM λ_m — is a 4 vector containing the spectral moments of order 0, 1, 2 and 4 in the first four addresses.

MD+ — is an *n* x 3 matrix, where the first column contains the threshold levels, and the second and third columns contain the corresponding ordinates of the PDF and CDF, respectively.

R=r1,r2 — are the lower and upper threshold bounds expressed in terms of units of the root-mean-square, i.e., square root of the spectral moment of order 0. (Default: *r1*=–1 and *r2*=4)

N=n — is the number of threshold levels to be considered. (Default: *N*=101)

The PDF and CDF † are respectively given by

$$f_p(x) = \frac{\sqrt{1-\alpha^2}}{\sqrt{2\pi\lambda_o}} exp\Big[-\frac{1}{2}\frac{x^2}{\lambda_o(1-\alpha^2)}\Big] + \frac{\alpha x}{\lambda_o} exp\Big(-\frac{x^2}{2\lambda_o}\Big)\Phi\Big(\frac{\alpha x}{\sqrt{(1-\alpha^2)\lambda_o}}\Big)$$

$$F_p(x) = \phi(\frac{x}{\sqrt{\lambda_0}\sqrt{1-\alpha^2}}) - \alpha\ exp(-\frac{x^2}{2\lambda_0})\ \phi(\frac{\alpha x}{\sqrt{(1-\alpha^2)\lambda_o}})$$

where $\alpha = \frac{\lambda_2}{\sqrt{\lambda_0\lambda_4}}$ and $\phi(.)$ and $\Phi(.)$ are the standard normal PDF and CDF, respectively.

† "The Statistical Distribution of Maxima of a Random Function," D. E. Cartwright and M. S. Longuet-Higgins, *Proceedings of the Royal Society of London*, Series A327, pp. 212-232, 1956.

MPF **MPHI VR VP+**

computes modal participation factors, where

MPHI is an n x m matrix containing the m mode shapes of an n-DOF system.

VR is an n vector of nodal load coefficients. For base input, use **VR** = $-M\ r$, where **M**= mass matrix and **r** = nodal displacements for unit base motion.

VP+ is an m vector containing the modal participation factors.

$$\mathbf{VP}(i) = \mathbf{MPHI}(k,i) * \mathbf{VR}(k) \qquad i = 1, ...m$$

NCFD **MRS MCD+** [*N=n IC=ic*]

constructs the normalized cumulative frequency diagram of a sample function contained in column *ic* of matrix **MRS**. This command can be used to examine the distribution of artificially generated random numbers, where

MRS is an *n* x *ns* matrix containing *ns* samples of size *n*.

MCD+ is a 2 or (*ns*+1) columns matrix, where the first column contains the equally spaced coordinates beginning at the smallest value and ending at the largest value of the sample, and the second column contains the corresponding ordinates of the normalized cumulative frequency diagram.

IC=ic is the column index of matrix **MRS** specifying the sample to be analyzed. All columns are analyzed if *ic* is not specified.

N=n is the number of equal width intervals used to compute the diagram. To determine the interval width, the difference between the largest and smallest data values is divided by the number of intervals *n*. (Default: *n*=10)

NCR **VR VCR+** [*X=xb, xe N=n*]

computes the mean upcrossing rates of a zero-mean nonstationary Gaussian process above specified thresholds, where

VR+	is an *nt* x 7 matrix, where the first column contains the time coordinates and the remaining six columns store the variances of the derivatives of order 0, 1 and 2, and the cross-correlation coefficients of the derivatives of orders 0 and 1, 0 and 2, and 1 and 2, respectively. Note that only the variances and cross-correlation coefficients of the first and second order derivatives are needed in the evaluation.
VCR+ $\nu(a^+,t)$	is an *nt* x (*n*+1) matrix, where the first column contains the time coordinates and the remaining *n* columns contain the mean up-crossing rates at the specified thresholds.
X=xb, xe a	are the lower and upper bounds of the equally spaced threshold (Default: $xb = xe = 0$)
N=n	is the number of threshold levels to be considered. (Default: n=1)

The mean upcrossing rate of level a becomes

$$\nu(a^+,t) = \frac{\sqrt{1-\rho^2_{x\dot{x}}(t)}}{\sqrt{2\pi}} \frac{\sigma_{\dot{x}}(t)}{\sigma_x(t)} \, exp\Big(-\frac{a^2}{2\sigma_x^2(t)}\Big) \, [\psi(r) + r\Phi(r)]$$

where $\psi(.)$ and $\Phi(.)$ denote the standard normal PDF PDF and CDF, respectively, and

$$r = \frac{\rho_{x\dot{x}}(t)}{\sqrt{1-\rho^2_{x\dot{x}}(t)}} \frac{a}{\sigma_x(t)}$$

NDEP **VR VEP+** *X=xb, xe* [*N=n*]

computes the PDF and CDF of the extreme peak of a zero-mean nonstationary Gaussian process, where

VR+ is an *nt* x 7 matrix, where the first column contains the time coordinates and the remaining six columns store the variances of the derivatives of order 0, 1 and 2, and the cross-correlation coefficients of the derivatives of orders 0 and 1, 0 and 2, and 1 and 2, respectively. Note that only the variances and cross-correlation coefficients of the first and second order derivatives are needed in the evaluation.

VEP+ is an *n* x 3 matrix, where the first column contains the specified threshold levels and the second and third columns contain the corresponding ordinates of the PDF and CDF of the extreme peak, respectively.

X=xb, xe are the lower and upper bounds of the specified threshold levels.

a

N=n is the number of threshold levels to be considered. (Default: n=1)

Assuming Poisson crossings, for the extreme peak defined as *max* $X(t)$, the CDF is given by

$$F_{x_T}(a) = exp[-\int_0^T \nu(a^+, t)dt]$$

where $\nu(a^+, t)$ is the mean upcrossing rate of level a. The PDF is computed by taking derivative with respect to a.

NDLP **VR VLP+** [*X=xb, xe N=n*]

computes the PDF of the local peaks for a zero-mean nonstationary Gaussian process, where

VR+ is an *nt* x 7 matrix, where the first column contains the time coordinates and the remaining six columns store the variances of the derivatives of order 0, 1 and 2, and the cross-correlation coefficients of the derivatives of orders 0 and 1, 0 and 2, and 1 and 2, respectively.

VLP+ is an *nt* x (*n*+1) matrix, where the first column contains the time coordinates and the remaining *n* columns contain the corresponding ordinates of the PDF of the local peaks.

X=xb, xe are the lower and upper bounds of the equally spaced threshold (Default: $xb = xe = 0$)

a

N=n is the number of threshold levels to be considered. (Default: n=1)

$$f_p(a;t) = \frac{\sqrt{\rho_0}}{[1-\rho^2_{x\dot{x}}(t)]\sqrt{1-\rho^2_{\dot{x}\ddot{x}}(t)}\sigma_x(t)} [\psi(\bar{r}) + \bar{r}\Phi(\bar{r})]$$
$$exp\left\{-\frac{1}{2\rho_0}(\frac{a}{\sigma_x(t)})^2\left[1-\rho^2_{\dot{x}\ddot{x}}(t) - \frac{[\rho_{x\ddot{x}}(t)-\rho_{x\dot{x}}(t)\rho_{\dot{x}\ddot{x}}(t)]^2}{[1-\rho^2_{x\dot{x}}(t)]^2}\right]\right\}$$

where $\psi(.)$ and $\Phi(.)$ denote the standard normal PDF PDF and CDF, respectively, and

$$\bar{r} = \frac{\rho_{x\ddot{x}}(t) - \rho_{x\dot{x}}(t)\rho_{\dot{x}\ddot{x}}(t)}{\sqrt{\rho_0}[1-\rho^2_{x\dot{x}}(t)]}\frac{a}{\sigma_x(t)} \qquad (2.122)$$

and $\rho_0 = 1 + 2\rho_{x\dot{x}}(t)\rho_{x\ddot{x}}(t)\rho_{\dot{x}\ddot{x}}(t) - \rho^2_{x\dot{x}}(t) - \rho^2_{x\ddot{x}}(t) - \rho^2_{\dot{x}\ddot{x}}(t)$.

NFD **MRS MFD+** [*N=n IC=ic*]

constructs the normalized frequency diagram of a sample function contained in column *ic* of matrix **MRS**. This command can be used to examine the distribution of artificially generated random numbers, where

MRS is an n x ns matrix containing ns samples of size n.

MCD+ is a 2 or $ns + 1$ columns matrix, where the first column contains the equally spaced coordinates beginning at the smallest value and ending at the largest value of the sample, and the second column contains the corresponding ordinates of the normalized frequency diagram.

IC=ic is the column index of matrix **MRS** specifying the sample to be analyzed. All columns are analyzed if *ic* is not specified.

N=n is the number of equal width intervals used to compute the diagram. To determine the interval width, the difference between the largest and smallest data values is divided by the number of intervals n. (Default: n=10)

PLOT **M1** [**M2** *N=n*] (*IX=ix1,ix2,...,ixn IY=iy1,iy2,...,iyn*)

plots on the console a graph with *n* lines based on x-y data in matrix **M1** (and matrix **M2**), where

M1 is a matrix in which the columns contains the x-coordinates when the second matrix **M2** is provided. When **M2** is not provided, the first column of **M2** specifies the x-coordinates and the remaining columns contain the y-coordinates.

M2 is a matrix with the same number of rows as **M1**, in which the columns contain the y-coordinates.

N=n is the total number of lines to be plotted. (Default: *n*=1)

IX=ix1,ix2,... is needed if two matrices are supplied, where *ixk* is the column index of matrix **M1** from which the x-coordinates of line *k* are chosen. (Default: *ix1*=1, *ix2*=2,...)

IY=iy1,iy2,... is needed if two matrices are supplied, where *iyk* is the column index of matrix **M2** from which the y-coordinates of line *k* are chosen. (Default: *iy1*=1, *iy2*=2,...)

PLOT SUB-COMMANDS

The sub-commands under *PLOT* are:

AXIS draws the x and y axes on the display and the plotter (if the plotter is toggled on.)

CLEAN cleans the buffer containing the plotting data.

RDATA reads the plotting data either from the database or from an external ASCII data file. The first row is the number of pairs of data used to draw the line and follows the coordinates for each point. And it may repeat for the data of next line.

ERASE erases the screen in order to draw a new graph.

HIDE hides the output text and message.

INIT initiates the parameters for line colors, line types, window location, etc.

PLOT toggles the plotter on or off. If it is toggled on, the *VIEW*, *AXIS*, *SCALE*, *TEXT* and *WIND* commands work on the display and on the plotter at the same time.

QUIT leaves *PLOT* (*SET*) sub-command level and goes back to STOCAL-II (*PLOT*) command level.

SCALE writes the minimum and maximum coordinates of the x and y data on the graph.

SET sets parameters for the location of the graph origin, the locations of the upper and lower corners of the plotting window, and the colors and types of the lines.

TEXT puts a string of text on the center bottom of the graph window.

VIEW plots the specified 2-D lines on the display and the plotter, if it is toggled on.

WIND draws a window for the graph.

ZOOM zooms the graph with respect to its origin.

SET SUB-COMMANDS

The sub-commands under *SET* are:

ORIGIN sets the origin point of the graph.

WINDOW sets the location of the graph window.

COLOR sets the colors of lines:
black=0; white=1; red=2; green=3; blue=4; yellow=5; cyan=6; magenta=7 (for IBM Enhanced Color Display)

TYPE sets the line types:
solid=1; Long dash=2; dotted=3; dash dotted=4; medium dashed=5; dash with two dots=6; short dash=7 (for IBM Enhanced Color Display)

HELP provides descriptions of the sub-commands.

RCQC **VW** **VD** **MZ** **VDT** **R+** [*IC=ic*]

computes the mean of absolute maximum of a response quantity by the CQC modal combination rule (Wilson et al. 1981, Der Kiureghian 1981), where

VW(ω_i)	is an n vector containing the modal frequencies in *rad/sec*.
VD(ς_i)	is an n vector containing the modal damping ratios.
MZ a_i	is an n rows matrix where the ic-th column contains the modal effective participation factors for the response quantity of interest.
VDT $\overline{D}_{ri}$	is an n vector containing the ordinates of the mean displacement response spectrum corresponding to the modal frequencies **VW** and damping ratios **VD**.
R+	is the mean of absolute maximum of the response quantity.
IC=ic	is the column index of matrix **MZ**, which contains the modal effective participation factors for the response quantity of interest. (Default: *ic=1*)

$$R = \Big(\sum_{i=1}^{n}\sum_{j=1}^{n} a_i a_j \;\; \rho_{0,ij} \;\; \overline{D}_{ri}\overline{D}_{rj}\Big)^{1/2}$$

$$\rho_{0,ij} = \frac{8\sqrt{\varsigma_i\varsigma_j}(\varsigma_i + r\varsigma_j)r^{3/2}}{(1-r^2)^2 + 4\varsigma_i\varsigma_j r(1+r^2) + 4(\varsigma_i^2+\varsigma_j^2)r^2}$$

$$r = \omega_j/\omega_i$$

(see Wilson et al. 1981†)

† "A Replacement for the SRSS Method in Seismic Analysis," E. L. Wilson, A. Der Kiureghian, and E. P. Bayo, *Earthquake Engineering and Structural Dynamics*, 9(5), pp. 187-194, 1981.

RSM **VW VD MZ VDT VM+ (MPI)** *T=τ I=type P=p1,p2,...* [*IC=ic*]

computes the spectral moments of order 0, 1, 2, and 4 for a response quantity of interest, when the input is specified by a mean displacement response spectrum. The combination rule uses a modal correlation coefficient matrix obtained for a stationary input with a PSD function of type *I*. Duration τ is used in computing peak factors, where

VW(ω_i) is an n vector containing the modal frequencies in *rad/sec*.

VD(ς_i) is an n vector containing the modal damping ratios.

MZ a_i is an n rows matrix where the *ic*-th column contains the modal effective participation factors for the response quantity of interest.

VDT $\overline{D}_{\tau i}$ is an n vector containing the ordinates of the mean displacement response spectrum corresponding to the modal frequencies **VW** and damping ratios **VD**.

VM+ λ_m is a 3 or 4 vector containing the spectral moments for the response quantity of interest.

MPI $\Phi_{ff}(\omega)$ is needed only for *I*=3 to specify the input PSD function used in computing the modal correlation coefficients. This is a two columns matrix, where the first column contains a sequence of non-negative, ascending frequency coordinates while the second column contains the corresponding ordinates of the PSD function. The function is assumed to be symmetric with respect to the frequency origin. Values outside the specified range are assumed to be zero.

I=type	*I*=1: White Noise
	I=2: Banded Linear Noise
	I=3: Arbitrary piecewise linear PSD specified by **MPI**.
	I=4: Filtered White Noise
P=p1,p2,...	are PSD parameters as follows:
	I=1: ***p1*** is the amplitude. (Default: ***p1***=1)
	I=2: ***p1*** is the amplitude at the lower frequency bound, ***p2*** is the lower frequency bound, ***p3*** is the upper frequency bound, and ***p4*** is the amplitude at the upper frequency bound. The function between the bounds is linear and outside the bounds is zero.
	I=3: ***p1*** is a scaler multiplier of **MPI**. (Default: ***p1***=1)
	I=4: ***p1*** (Φ_o) is the amplitude at zero frequency, and ***p2*** (ω_g) and ***p3*** (ς_g) are the natural frequency and damping ratio of the filter, respectively.
T=τ	is the equivalent duration of the stationary phase.
IC=ic	is the column index of matrix **MZ**, which contains the modal effective participation factors for the response quantity of interest. (Default: *ic=1*)

$$\lambda_m = \sum_{i=1}^{n}\sum_{j=1}^{n} a_i a_j \;\rho_{m,ij}\;\omega_{m,i}\omega_{m,j}\frac{1}{p_i p_j}\overline{D}_{\tau i}\overline{D}_{\tau j}$$

where $\rho_{m,ij} = \frac{\lambda_{m,ij}}{\sqrt{\lambda_{m,ii}\lambda_{m,jj}}}$, $\omega_{m,i} = \sqrt{\frac{\lambda_{m,ii}}{\lambda_{0,ii}}}$ and $p_i = p_i(\nu_i(0)\tau, \delta_i)$. $\rho_{m,ij}$, $\omega_{m,i}$ and p_i are computed based on the specified PSD function (see Der Kiureghian 1981†).

† "A Response Spectrum Method for Random Vibration Analysis of MDOF Systems," A. Der Kiureghian, *Earthquake Engineering and Structural Dynamics*, 9, pp. 419-435. 1981.

SCF **VW VD MZ1 MZ2 MCF+ (MPI)** *I=type* *P=p1,p2,...* [*TA=tb,te* \
N=nt *M=m1,m2* *IC=i1,i2* *L=l*]

computes the stationary auto or cross-correlation function of response quantities Z1($t1$) and Z2($t2$), or their time derivatives, for an input excitation specified by a PSD function of type *I*. Autocorrelation is computed when Z1 and Z2 denote the same response quantity; otherwise, cross-correlation is computed, where

VW(ω_i) is an n vector containing the modal frequencies in *rad/sec*.

VD(ζ_i) is an n vector containing the modal damping ratios.

MZ1 a_{1i} is an n rows matrix, where the *i1*-th column contains the modal effective participation factors for the response Z1.

MZ2 a_{2j} is an n rows matrix, where the *i2*-th column contains the modal effective participation factors for the response Z2.

MCF+ $\phi_{z_1,z_2}^{(m_1,m_2)}(\tau)$ is an *nt* x 3 matrix, where the first column contains a sequence of equally spaced time-lag coordinates beginning at *tb* and ending at *te*, while the second column contains the corresponding auto or cross-correlation values.

MPI $\Phi_{ff}(\omega)$ is needed only for *I*=3 to specify the input PSD function. This is a two or three columns matrix, where the first column contains a sequence of non-negative, ascending frequency coordinates, while the second and third columns contain the corresponding real and imaginary parts of the PSD function. If the imaginary part is zero, only two columns are needed. The real part of **MPI** is assumed to be an even function while the imaginary part is assumed to be an odd function. The PSD is assumed to be zero outside the specified frequency interval.

I=type I=1: White Noise

I=2: Banded Linear Noise

I=3: Arbitrary piecewise linear PSD specified by **MPI**.

I=4: Filtered White Noise

P=p1,p2,... are PSD parameters as follows:

I=1: ***p1*** is the amplitude. (Default: ***p1***=1)

I=2: ***p1*** is the amplitude at the lower frequency bound, ***p2*** is the lower frequency bound, ***p3*** is the upper frequency bound, and ***p4*** is the amplitude at the upper frequency bound. The function between the bounds is linear and outside the bounds is zero.

I=3: ***p1*** is a scaler multiplier of **MPI**. (Default: ***p1***=1)

I=4: ***p1*** (Φ_o) is the amplitude at zero frequency, and ***p2*** (ω_g) and ***p3*** (ς_g) are the natural frequency and damping ratio of the filter, respectively.

TA=tb,te are the lower and upper time-lag limits of the output data in *sec*. (Default: ***tb***=0, ***te***=10)

N=nt is the number of time-lag points between ***tb*** and ***te***. (Default: ***nt***=101).

M=m1,m2 are the derivative orders for the two response quantities, respectively. (Default: ***m1***=***m2***=0)

IC=i1,i2 are the column indices of matrices **MZ1** and **MZ2**, which contain the modal effective participation factors for response Z1 and Z2, respectively. (Default: ***i1***=***i2***=1)

L=l is the number of modes considered. (Default: l=n)

$$R_{x_1,x_2}^{(m_1,m_2)}(\tau) = \sum_{i=1}^{l}\sum_{j=1}^{l} \; a_{1i} \; a_{2j} \; R_{ij}^{(m_1,m_2)}(\tau) \qquad t_b \le \tau \le t_e$$

$$R_{ij}^{(m_1,m_2)}(\tau) = \int_{-\infty}^{\infty} \; (\mathrm{i}\omega)^{m_1}(-\mathrm{i}\omega)^{m_2} H_i(\omega_i) H_j^*(\omega_j) \;\; \Phi_{ff}(\omega) e^{\mathrm{i}\omega\tau} d\omega$$

where $H_i(\omega_i)$ is the frequency response function of mode i.

SM **MPI** **VSM+** [*P*=*p*]

computes the spectral moments of order 0, 1, 2, and 4 for a specified PSD function, where

MPI $\Phi(\omega)$	is a two columns matrix where the first column contains a sequence of non-negative, ascending frequency coordinates, while the second column contains the corresponding ordinates of the PSD function.
VSM+ λ_m	is a 4 vector containing the spectral moments of order 0, 1, 2 and 4 in sequence.
P=*p*	is a scaler multiplier of function **MPI**. (Default: *p*=1)

$$\lambda_m = 2p\int_0^\infty \omega^m \Phi(\omega)d\omega$$

or for a piecewise linear PSD function

$$\lambda_m = 2p\sum_{k=1}^{n-1} \frac{a_k}{m+2}[(\omega_{k+1})^{m+2} - (\omega_k)^{m+2}] + \frac{b_k}{m+1}[(\omega_{k+1})^{m+1} - (\omega_k)^{m+1}]$$

where m = 0, 1 ,2 or 4, n is the total number of rows in matrix **MPI** and

$$a_k = \frac{\Phi(\omega_{k+1}) - \Phi(\omega_k)}{\omega_{k+1} - \omega_k} \quad \text{and} \quad b_k = \Phi(\omega_k) - a_k\omega_k$$

SMR **VMSM MCC MZ1 MZ2 VSM+** [*IC=i1,i2*]

computes the m-th spectral moment of response quantities Z1 and Z2 using modal spectral moments provided in matrices **VMSM** and **MCC**, which may be formed by using command *SMSM*, where

VMSM (λ_m) is an n vector containing the m-th modal spectral moments.

MCC $\rho_{m,ij}$ is an n x n correlation coefficient matrix associated with the cross-modal spectral moments of order m.

MZ1(a_{1i}) is an n rows matrix where the *i1*-th column contains the modal effective participation factors for the response Z1.

MZ2(a_{2j}) is an n rows matrix where the *i2*-th column contains the modal effective participation factors for the response Z2.

VSM+ contains the m-th response spectral moment.

IC=i1,i2 are the column indices of matrices **MZ1** and **MZ2**, which contain the modal effective participation factors for the response Z1 and Z2, respectively. (Default: *i1*=*i2*=1)

$$VSM = \sum_{i}^{n}\sum_{j}^{n} a_{1i}\ a_{2j}\ \rho_{m,ij}\ \sqrt{\lambda_{m,ii}}\ \sqrt{\lambda_{m,jj}}$$

SMSM **VW VD VMSM+ MCC+ (MPI)** *I=type P=p1,p2,...* [*M=m L=l*]

computes the *m*-th modal spectral moments and the associated correlation coefficient matrix an input excitation specified by a PSD function of type *I*, where

VW(ω_i)	is an *n* vector containing the modal frequencies in *rad/sec*.
VD(ς_i)	is an *n* vector containing the modal damping ratios.
VMSM+(λ_m)	is an *l* vector containing the *m*-th modal spectral moments.
MCC+ $\rho_{m,ij}$	is an *l* x *l* correlation coefficient matrix associated with the cross-modal spectral moments of order *m*.
MPI $\Phi_{ff}(\omega)$	is needed only for *I*=3 to specify the input PSD function. This is a two or three columns matrix, where the first column contains a sequence of non-negative, ascending frequency coordinates, while the second and third columns contain the corresponding real and imaginary parts of the PSD function. If the imaginary part is zero, only two columns are needed. The real part of **MPI** is assumed to be an even function while the imaginary part is assumed to be an odd function. The PSD is assumed to be zero outside the specified frequency interval.
I=type	*I*=1: White Noise *I*=2: Banded Linear Noise *I*=3: Arbitrary piecewise linear PSD specified by **MPI**. *I*=4: Filtered White Noise
P=p1,p2,...	are PSD parameters as follows: *I*=1: *p1* is the amplitude. (Default: *p1*=1) *I*=2: *p1* is the amplitude at the lower frequency bound, *p2* is the lower frequency bound, *p3* is the upper frequency bound, and

p4 is the amplitude at the upper frequency bound. The function between the bounds is linear and outside the bounds is zero.

I=3: *p1* is a scaler multiplier of **MPI**. (Default: *p1*=1)

I=4: *p1* (Φ_o) is the amplitude at zero frequency, and *p2* (ω_g) and *p3* (ς_g) are the natural frequency and damping ratio of the filter, respectively.

M=*m* is the power of the frequency multiplier of the PSD function. (Default: *m*=0)

L=*l* is the number of modes considered. (Default: *l*=n)

$$\lambda_{m,ij} = 2 \; Re \int_0^\infty (\omega)^m H_i(\omega_i) H_j^*(\omega_j) \;\; \Phi_{ff}(\omega) d\omega$$

and

$$\rho_{m,ij} = \frac{\lambda_{m,ij}}{\sqrt{\lambda_{m,ii}\lambda_{m,jj}}}$$

where $H_i(\omega_i)$ is the frequency response function of mode i.

SPSD **VW VD MZ1 MZ2 MPO+ (MPI)** *I=type P=p1,p2,...* [*W=wb,we* \
N=nw M=m1,m2 IC=i1,i2 L=l]

computes the stationary auto or cross-PSD function of response quantities Z1(t) and Z2(t), or their time derivatives for the input excitation specified by a PSD function of type *I*. Auto-PSD is computed when Z1 and Z2 denote the same response quantity; otherwise, cross-PSD is computed, where

VW(ω_i) is an n vector containing the modal frequencies in *rad/sec*.

VD(ς_i) is an n vector containing the modal damping ratios.

MZ1 a_{1i} is an n rows matrix, where the *i1*-th column contains the modal effective participation factors for the response Z1.

MZ2 a_{2j} is an n rows matrix, where the *i2*-th column contains the modal effective participation factors for the response Z2.

MPO+ $\Phi_{z_1,z_2}^{(m_1,m_2)}(\omega)$ is an nw x 3 matrix, where the first column of the matrix contains a sequence of equally spaced frequency coordinates beginning at wb and ending at we, while the second and third columns contain the corresponding real and imaginary parts of the resulting PSD function.

MPI $\Phi_{ff}(\omega)$ is needed only for *I*=3 to specify the input PSD function. This is a two or three columns matrix, where the first column contains a sequence of non-negative, ascending frequency coordinates, while the second and third columns contain the corresponding real and imaginary parts of the PSD function. If the imaginary part is zero, only two columns are needed. The real part of **MPI** is assumed to be an even function while the imaginary part is assumed to be an odd function. The PSD is assumed to be zero outside the specified frequency interval.

I=type — *I*=1: White Noise

I=2: Banded Linear Noise

I=3: Arbitrary piecewise linear PSD specified by **MPI**.

I=4: Filtered White Noise

P=p1,p2,... — are PSD parameters as follows:

I=1: *p1* is the amplitude. (Default: *p1*=1)

I=2: *p1* is the amplitude at the lower frequency bound, *p2* is the lower frequency bound, *p3* is the upper frequency bound, and *p4* is the amplitude at the upper frequency bound. The function between the bounds is linear and outside the bounds is zero.

I=3: *p1* is a scaler multiplier of **MPI**. (Default: *p1*=1)

I=4: *p1* (Φ_o) is the amplitude at zero frequency, and *p2* (ω_g) and *p3* (ς_g) are the natural frequency and damping ratio of the filter, respectively.

W=wb,we — are the lower and upper frequency limits of the output data in *rad/sec*. (Default: *wb*=0, *we*=100)

N=nw — is the number of points to be generated for the output data. (Default: *nw* = 101).

M=m1,m2 — are the derivative orders for the two response quantities, respectively. (Default: *m1*=*2*=0)

IC=i1,i2 — are the column indices of matrices **MZ1** and **MZ2**, which contain the modal effective participation factors for response Z1 and Z2, respectively. (Default: *i1*=*i2*=1)

L=l — is the number of modes considered. (Default: *l*=n)

$$\Phi_{z_1,z_2}^{(m_1,m_2)}(\omega) = \sum_{i=1}^{l}\sum_{j=1}^{l} a_{1i} \ a_{2j} \ \Phi_{ij}^{(m_1,m_2)}(\omega) \qquad w_b \le \omega \le w_e$$

where

$$\Phi_{ij}^{(m_1,m_2)}(\omega) = (\mathrm{i}\omega)^{m_1}(-\mathrm{i}\omega)^{m_2} H_i(\omega_i) H_j^*(\omega_j) \ \Phi_{ff}(\omega)$$

where $H_i(\omega_i)$ is the frequency response function of mode i.

SRSM **VW VD MZ1 MZ2 VSM+ (MPI)** *I=type P=p1,p2,...* [*IC=i1,i2 L=l*]

computes the auto or cross-response spectral moments of order 0, 1, 2 and 4, of response components Z1(t) and Z2(t) for an input excitation specified by a PSD function of type I. Auto-RSM is computed when Z1 and Z2 denote the same response quantity; otherwise, cross-RSM is computed, where

VW(ω_i) is an n vector containing the modal frequencies in *rad/sec*.

VD(ς_i) is an n vector containing the modal damping ratios.

MZ1 a_{1i} is an n rows matrix, where the *i1*-th column contains the modal effective participation factors for the response Z1.

MZ2 a_{2j} is an n rows matrix, where the *i2*-th column contains the modal effective participation factors for the response Z2.

VSM+ λ_m is a 4 vector containing the spectral moments of order 0, 1, 2 and 4 in sequence.

MPI $\Phi_{ff}(\omega)$ is needed only for *I*=3 to specify the input PSD function. This is a two or three columns matrix, where the first column contains a sequence of non-negative, ascending frequency coordinates, while the second and third columns contain the corresponding real and imaginary parts of the PSD function. If the imaginary part is zero, only two columns are needed. The real part of **MPI** is assumed to be an even function while the imaginary part is assumed to be an odd function. The PSD is assumed to be zero outside the specified frequency interval.

I=type — I=1: White Noise

I=2: Banded Linear Noise

I=3: Arbitrary piecewise linear PSD specified by **MPI**.

I=4: Filtered White Noise

P=p1,p2,... — are PSD parameters as follows:

I=1: *p1* is the amplitude. (Default: *p1*=1)

I=2: *p1* is the amplitude at the lower frequency bound, *p2* is the lower frequency bound, *p3* is the upper frequency bound, and *p4* is the amplitude at the upper frequency bound. The function between the bounds is linear and outside the bounds is zero.

I=3: *p1* is a scaler multiplier of **MPI**. (Default: *p1*=1)

I=4: *p1* (Φ_o) is the amplitude at zero frequency, and *p2* (ω_g) and *p3* (ς_g) are the natural frequency and damping ratio of the filter, respectively.

IC=i1,i2 — are the column indices of matrices **MZ1** and **MZ2**, which contain the modal effective participation factors for response Z1 and Z2, respectively. (Default: *i1*=*i2*=1)

L=l — is the number of modes considered. (Default: l=n)

$$\lambda_m = \sum_{i=1}^{l} \sum_{j=1}^{l} a_{1i} \; a_{2j} \; \lambda_{m,ij}$$

where

$$\lambda_{m,ij} = 2 \; Re \int_0^\infty (\omega)^m H_i(\omega_i) H_j^*(\omega_j) \; \Phi_{ff}(\omega) d\omega$$

and m = 0, 1, 2, and 4.

SSGP **VSM** $(T{=}\tau)$ $[X{=}xb, xe\ \ N{=}n\ \ MU{=}\mu_X]$

computes the statistics of a stationary Gaussian process X(t) with mean μ_X. These include the root mean squares of X(t), and its first and second derivatives, mean upcrossing rates, statistics of the envelope process, and PSD and CDF of the process, its envelope and local and extreme peaks, where

VSM	is a 4 vector containing the spectral moments of order 0, 1, 2 and 4 in sequence.
$T{=}\tau$	is the duration of the process; only needed if the extreme peak is required.
$X{=}xb, xe$	are the lower and upper bounds of the threshold levels. (Default: $X{=}\mu_X$)
$N{=}n$	is the number of equally spaced threshold levels. (Default: $n{=}1$)
$MU{=}\mu_X$	is the mean value of the process. (Default: $\mu_X{=}0$)

STAT **MRN MST+**

computes the means, standard deviations and skewness coefficients of *ns* random samples of size *n* contained in matrix **MRN**, where

MRN is an *n* x *ns* matrix containing *ns* sets of *n* samples.

MST+ is a 3 x *ns* matrix containing the means, standard deviations, and skewness coefficients.

$$\mathbf{MST}(1,j) = \frac{1}{n}\sum_{k=1}^{ns}\mathbf{MRN}(k,j)$$

$$\mathbf{MST}(2,j) = \left[\frac{1}{n}\sum_{k=1}^{ns}\mathbf{MRN}(k,j)^2 - \mathbf{MST}(1,j)^2\right]^{1/2}$$

$$\mathbf{MST}(3,j) = \frac{1}{n * \mathbf{MST}(2,j)^3}\sum_{k=1}^{ns}[\mathbf{MRN}(k,j) - \mathbf{MST}(1,j)]^3$$

TACF **MX MCF+** *DT=dt* [*IR=ir*]

computes the temporal autocorrelation function (second order temporal average) of the sample function contained in row *ir* of matrix MX, where

MX	**is an n x nt matrix containing n sets of random sample functions each specified at nt points.**
MCF+	**is a two or $n+1$ columns matrix, where the first column contains the time-lag coordinates while the remaining columns contain the temporal autocorrelation values.**
DT=dt	**is the time increment.**
IR=ir	**is the row index specifying the row to be analyzed. (Default: all rows are analyzed.)**

$$\mathbf{MCF}(i,1) = (i-1) * dt$$

$$\mathbf{MCF}(i,j) = \sum_{k=1}^{nt-i+1} \frac{1}{nt-i+1} \mathbf{MX}(ir,k) * \mathbf{MX}(ir,k+i-1) \quad j = 1,2,...,\mathrm{int}(\frac{nt}{4})$$

TCF **VW VD MZ1 MZ2 MT MCF+ (MPI)** *I=type P=p1,p2,...* \
T1=t1b,t1e [*T2=t2b,t2e N=nt M=m1,m2 IC=i1,i2 L=l*]

computes the auto or cross-correlation function of response quantities Z1(t1) and Z2(t2) or their time derivatives for an input excitation specified by a uniformly modulated PSD function, i.e., a PSD function of type *I* modulated by the time function **MT**, where

VW(ω_i)	is an *n* vector containing the modal frequencies in *rad/sec*.
VD(ζ_i)	is an *n* vector containing the modal damping ratios.
MZ1 a_{1i}	is an *n* rows matrix, where the *i1*-th column contains the modal effective participation factors for the response Z1.
MZ2 a_{2j}	is an *n* rows matrix, where the *i2*-th column contains the modal effective participation factors for the response Z2.
MT $A(\tau)$	is a two columns matrix describing the time modulating function for the input process, where the first column contains a sequence of ascending time coordinates and the second column contains the corresponding ordinates of the modulating function. The ordinates before the first time coordinate are assumed to be zero, while the ordinates after the last time coordinate are assumed to remain the same as the last ordinate.
MCF+ $\phi_{z_1,z_2}^{(m_1,m_2)}(t_1,t_2)$	is an *nt* x 3 matrix, where the first and second columns contain the time coordinates *t1* and *t2*, respectively, and the third column contains the corresponding auto or cross-correlation values.

MPI $\Phi_{ff}(\omega)$	is needed only for *I*=3 to specify the input PSD function. This is a two or three columns matrix, where the first column contains a sequence of non-negative, ascending frequency coordinates while the second and third columns contain the corresponding real and imaginary parts of the PSD function. If the imaginary part is zero, only two columns are needed. The real part of **MPI** is assumed to be an even function while the imaginary part is assumed to be an odd function. The PSD is assumed to be zero outside the specified frequency interval.
I=type	*I*=1: White Noise *I*=2: Banded Linear Noise *I*=3: Arbitrary piecewise linear PSD specified by **MPI**.
P=p1,p2,...	are PSD parameters as follows: *I*=1: *p1* is the amplitude. (Default: *p1*=1) *I*=2: *p1* is the amplitude at the lower frequency bound, *p2* is the lower frequency bound, *p3* is the upper frequency bound, and *p4* is the amplitude at the upper frequency bound. The function between the bounds is linear and outside the bounds is zero. *I*=3: *p1* is a scaler multiplier of **MPI**. (Default: *p1*=1)
T1=t1b,t1e	are the lower and upper bounds for time coordinates *t*1 associated with the first response quantity.
T2=t2b,t2e	are the lower and upper bounds for time coordinates *t*2 associated with the second response quantity. (Default: *T2=T1*)
N=nt	is the number of time points between *t1b* and *t1e* and between *t2b* and *t2e*. (Default: *nt*=2)

M=m1,m2 are the derivative orders for the two response quantities, respectively. (Default: *m1=m2*=0)

IC=i1,i2 are the column indices of matrices **MZ1** and **MZ2**, which contain the modal effective participation factors for response Z1 and Z2, respectively. (Default: *i1=1=i2*=1)

L=l is the number of modes considered. (Default: *l*=n)

$$\phi_{z_1,z_2}^{(m_1,m_2)}(t_1,t_2) = \sum_{i=1}^{l}\sum_{j=1}^{l} a_{1i}\ a_{2j}\ \phi_{ij}^{(m_1,m_2)}(t_1,t_2)$$

where

$$\phi_{ij}^{(m_1,m_2)}(t_1,t_2) = \int_{-\infty}^{\infty} \frac{\partial^{m_1}[\int_0^{t_1} A(\tau)h_i(t_1-\tau)e^{i\omega\tau}d\tau]}{\partial t_1^{m_1}} \frac{\partial^{m_2}[\int_0^{t_2} A(\tau)h_j(t_2-\tau)e^{-i\omega\tau}d\tau]}{\partial t2^{m_2}} \Phi_{ff}(\omega)d\omega$$

where $h_i(t)$ is the unit impulse response function of mode i.

TFSU **MSU MRS+** *I=type* *P=p1,p2*

transforms matrix **MSU** containing random samples with the standard uniform distribution into samples with a specified distribution, where

MSU(z) is an n x ns matrix containing ns sets of n random values with uniform distribution between 0 and 1. This matrix can be generated by using command *GSU*.

MRS+(x) is an n x ns matrix containing the transformed samples with the specified distribution.

P=p1,p2 are the set of parameters of the specified distribution.

I=type *I*=1: uniform distribution within [*p1, p2*]

$$x = (p2 - p1) * z + p1$$

where $f(x) = 1/(p2 - p1), \quad p1 < x < p2$

I=2: exponential distribution with mean *p1* ($p1 > 0$)

$$x = -ln(1 - z)/p1$$

where $f(x) = p1 \; e^{-p1*x}, \quad p1 < x < p2$

I=3: normal distribution with mean *p1* and standard deviation *p2* ($p2 > 0$)

$$x = p1 + p2 * \Phi^{-1}(z)$$

where Φ^{-1} is for the inverse of the standard normal CDF, and

$$f(x) = \frac{1}{\sqrt{2\pi}\;p2} exp[-\frac{(x-p1)^2}{2\;p2^2}]$$

I=4: lognormal distribution with parameters *p1* and *p2* ($p1, \; p2 > 0$)

$$x = exp[p1 + p2 * \Phi^{-1}(z)]$$

$$f(x) = \frac{1}{\sqrt{2\pi}\;p2} exp\{-\frac{[ln(x)-p1]^2}{2\;p2^2}\}$$

TMS **MT VR+ (MPI)** *I=type P=p1,p2,... T=tb,te* [*N=nt*]

computes the variances and cross-correlation coefficients of a uniformly modulated process and its derivatives. The modulated process is described by a uniformly modulated PSD function, i.e., a stationary process of type *I* modulated by the time function **MT**, where

MT $A(t)$ — is a two columns matrix describing the time modulating function for the input process, where the first column contains a sequence of ascending time coordinates and the second column contains the corresponding ordinates of the modulating function. The ordinates before the first time coordinate are assumed to be zero, while the ordinates after the last time coordinate are assumed to remain the same as the last ordinate.

VR+ — is an *nt* x 7 matrix, where the first column contains the time coordinates and the remaining six columns store the variances of the derivatives of order 0, 1 and 2, and the cross-correlation coefficients of the derivatives of orders 0 and 1, 0 and 2, and 1 and 2, respectively.

MPI $\Phi(\omega)$ — is needed only for *I*=3 to specify the input PSD function. This is a two or three columns matrix, where the first column contains a sequence of non-negative, ascending frequency coordinates while the second and third columns contain the corresponding real and imaginary parts of the PSD function. If the imaginary part is zero, only two columns are needed. The real part of **MPI** is assumed to be an even function while the imaginary part is assumed to be an odd function. The PSD is assumed to be zero outside the specified frequency interval.

I=type	*I*=2: Banded Linear Noise
	I=3: Arbitrary piecewise linear PSD specified by **MPI**.
P=p1,p2,...	are PSD parameters as follows:
	I=2: *p1* is the amplitude at the lower frequency bound, *p2* is the lower frequency bound, *p3* is the upper frequency bound, and *p4* is the amplitude at the upper frequency bound. The function between the bounds is linear and outside the bounds is zero.
	I=3: *p1* is a scaler multiplier of **MPI**. (Default: *p1*=1)
T=tb,te	are the lower and upper bounds of equally spaced time coordinates.
N=nt	is the number of time coordinates. (Default: *nt*=2)

$$\phi_{xx}^{(m1,m2)}(t,t) = \int_{-\infty}^{\infty} \frac{\partial^{m1} A(t)e^{i\omega t}}{\partial t^{m1}} \frac{\partial^{m2} A(t)e^{-i\omega t}}{\partial t^{m2}} \Phi(\omega)d\omega$$

TPSD **VW** **VD** **MZ1** **MZ2** **MT** **MPO+** **(MPI)** *I=type* *P=p1,p2,...* \
[*W=wb,we* *N=nw*] *T1=t11,t12,...* [*T2=t21,t22,...* *M=m1,m2* *IC=i1,i2* *L=l*]

computes the evolutionary auto or cross-PSD function of response quantities Z1(t1) and Z2(t2), or their time derivatives, for an input excitation specified by a uniformly modulated PSD function, i.e., a PSD function of type *I* modulated by time function **MT**, where

VW(ω_i)	is an *n* vector containing the modal frequencies in *rad/sec*.
VD(ζ_i)	is an *n* vector containing the modal damping ratios.
MZ1 a_{1i}	is an *n* rows matrix, where the *i1*-th column contains the modal effective participation factors for the response Z1.
MZ2 a_{2j}	is an *n* rows matrix, where the *i2*-th column contains the modal effective participation factors for the response Z2.
MT $A(\tau)$	is a two columns matrix describing the time modulating function for the input process, where the first column contains a sequence of ascending time coordinates and the second column contains the corresponding ordinates of the modulating function. The ordinates before the first time coordinate are assumed to be zero, while the ordinates after the last time coordinate are assumed to remain the same as the last ordinate.
MPO+ $\Phi_{z_1,z_2}^{(m_1,m_2)}(\omega,t_1,t_2)$	is an *nw* rows matrix, where the first column of the matrix contains a sequence of *nw* equally spaced frequency coordinates beginning at *wb* and ending at *we*, while the 2nd, 4th,... columns contain the real parts and the 3rd, 5th,... columns contain the imaginary parts of the resulting PSD functions at the specified times.

MPI $\Phi_{ff}(\omega)$	is needed only for *I*=3 to specify the input PSD function. This is a two or three columns matrix, where the first column contains a sequence of non-negative, ascending frequency coordinates, while the second and third columns contain the corresponding real and imaginary parts of the PSD function. If the imaginary part is zero, only two columns are needed. The real part of **MPI** is assumed to be an even function while the imaginary part is assumed to be an odd function. The PSD is assumed to be zero outside the specified frequency interval.
I=type	*I*=1: White Noise *I*=2: Banded Linear Noise *I*=3: Arbitrary piecewise linear PSD specified by **MPI**. *I*=4: Filtered White Noise
P=p1,p2,...	are PSD parameters as follows: *I*=1: ***p1*** is the amplitude. (Default: ***p1***=1) *I*=2: ***p1*** is the amplitude at the lower frequency bound, ***p2*** is the lower frequency bound, ***p3*** is the upper frequency bound, and ***p4*** is the amplitude at the upper frequency bound. The function between the bounds is linear and outside the bounds is zero. *I*=3: ***p1*** is a scaler multiplier of **MPI**. (Default: ***p1***=1) *I*=4: ***p1*** (Φ_o) is the amplitude at zero frequency, and ***p2*** (ω_g) and ***p3*** (ς_g) are the natural frequency and damping ratio of the filter, respectively.
W=wb,we	are the lower and upper frequency limits of the output data in *rad/sec*. (Default: ***wb***=0, ***we***=100)

N=nw is the number of points to be generated for the output data. (Default: $nw = 101$).

T1=t11,t12,... are the time points of interest for the first response quantity.

T2=t21,t22,... are the time points of interest for the second response quantity. (Default: *T2=T1*)

M=m1,m2 are the derivative orders for the two response quantities, respectively. (Default: *m1=2=0*)

IC=i1,i2 are the column indices of matrices **MZ1** and **MZ2**, which contain the modal effective participation factors for response Z1 and Z2, respectively. (Default: *i1=i2=1*)

L=l is the number of modes considered. (Default: *l*=n)

$$\Phi_{z_1,z_2}^{(m_1,m_2)}(\omega,t_1,t_2) = \sum_{i=1}^{l}\sum_{j=1}^{l} a_{1i}\ a_{2j}\ \Phi_{ij}^{(m_1,m_2)}(\omega,t_1,t_2)$$

where

$$\Phi_{ij}^{(m_1,m_2)}(\omega,t_1,t_2) = \frac{\partial^{m_1}[\int_0^{t_1} A(\tau)h_j(t_1-\tau)e^{i\omega\tau}d\tau]}{\partial t_1^{m_1}} \frac{\partial^{m_2}[\int_0^{t_2} A(\tau)h_k(t_2-\tau)e^{-i\omega\tau}d\tau]}{\partial t_2^{m_2}}\ \Phi_{ff}(\omega)e^{i\omega(t_2-t_1)}$$

where $h_i(t)$ is the unit impulse response function of mode i.

TRMS **VW VD MZ MT VR+ (MPI)** *I=type P=p1,p2,... T=tb,te* \
[N=nt IC=ic L=l]

computes the variances and cross-correlation coefficients of the response $Z(t)$ and its first and second derivatives for an input excitation specified by a uniformly modulated PSD function, i.e., a PSD function of type *I* modulated by time function **MT**, where

VW(ω_i) is an *n* vector containing the modal frequencies in *rad/sec*.

VD(ς_i) is an *n* vector containing the modal damping ratios.

MZ
a_i is an *n* rows matrix where the *ic*-th column contains the modal effective participation factors for the response quantity of interest.

MT
$A(\tau)$ is a two columns matrix describing the time modulating function for the input process, where the first column contains a sequence of ascending time coordinates and the second column contains the corresponding ordinates of the modulating function. The ordinates before the first time coordinate are assumed to be zero, while the ordinates after the last time coordinate are assumed to remain the same as the last ordinate.

VR+ is an *nt* x 7 matrix, where the first column contains the time coordinates and the remaining six columns store the variances of the derivatives of order 0, 1 and 2, and the cross-correlation coefficients of the derivatives of orders 0 and 1, 0 and 2, and 1 and 2, respectively.

MPI
$\Phi_{ff}(\omega)$ is needed only for *I*=3 to specify the input PSD function. This is a two or three columns matrix, where the first column contains a sequence of non-negative, ascending frequency coordinates while the

second and third columns contain the corresponding real and imaginary parts of the PSD function. If the imaginary part is zero, only two columns are needed. The real part of **MPI** is assumed to be an even function while the imaginary part is assumed to be an odd function. The PSD is assumed to be zero outside the specified frequency interval.

I=type   *I*=1: White Noise

I=2: Banded Linear Noise

I=3: Arbitrary piecewise linear PSD specified by **MPI**.

P=p1,p2,...   are PSD parameters as follows:

I=1: ***p1*** is the amplitude. (Default: ***p1***=1)

I=2: ***p1*** is the amplitude at the lower frequency bound, ***p2*** is the lower frequency bound, ***p3*** is the upper frequency bound, and ***p4*** is the amplitude at the upper frequency bound. The function between the bounds is linear and outside the bounds is zero.

I=3: ***p1*** is a scaler multiplier of **MPI**. (Default: ***p1***=1)

T=tb,te   are the lower and upper bounds of equally spaced time coordinates.

N=nt   is the number of time coordinates. (Default: ***nt***=2)

IC=ic   is the column index of matrix **MZ**, which contains the modal effective participation factors for the response quantity of interest. (Default: *ic=1*)

L=l   is the number of modes considered. (Default: *l*=n)

$$\phi_{z_1,z_2}^{(m_1,m_2)}(t,t) = \sum_{i}^{l}\sum_{j}^{l} a_{1i} \ a_{2j} \ \phi_{ij}^{(m_1,m_2)}(t,t)$$

where

$$\phi_{ij}^{(m_1,m_2)}(t,t) = \int_{-\infty}^{\infty} \frac{\partial^{m_1}[\int_0^t A(\tau)h_i(t-\tau)e^{i\omega\tau}d\tau]}{\partial t^{m_1}} \frac{\partial^{m_2}[\int_0^t A(\tau)h_j(t-\tau)e^{-i\omega\tau}d\tau]}{\partial t^{m_2}} \Phi_{ff}(\omega)d\omega$$

where $h_i(t)$ is the unit impulse response function of mode i.

TSSF **MX MA MT+** *T=tb,te*

multiplies generated sample functions contained in matrix **MX** by the time modulating function **MT**, where

MX is an *n* x *nt* matrix containing *n* sample functions specified at *nt* equally spaced points. The first column represents the samples at time *tb* and the last column represents the samples at time *te*.

MT is a two columns matrix, where the first column contains a sequence of ascending time coordinates and the second column contains the corresponding ordinates of the modulating function.

MY+ is the resulting *n* x *nt* matrix obtaining the modulating sample functions.

T=tb,te are the time coordinates for the first and the last columns of matrix **MX**.

TTSU **MRS** **MSU+** *I=type P=p1,p2*

transforms matrix MRS containing random samples with a specified distribution into a new matrix MSU containing samples with the standard uniform distribution, where

MRS(x)	is an n x ns matrix containing ns sets of n random values with the specified distribution.
MSU+(z)	is an n x ns matrix containing the transformed samples with uniform distribution between 0 and 1.
P=p1,p2	are the set of parameters of the specified distribution.
I=type	*I*=1: uniform distribution within [*p1, p2*]

$$z = (x - p1)/(p2 - p1)$$

where $f(x) = 1/(p2 - p1) \qquad p1 < x < p2$

I=2: exponential distribution with mean *p1* ($p1 > 0$)

$$z = 1 - e^{-p1*x}$$

where $f(x) = p1 \;\; e^{-p1*x}, \qquad p1 < x < p2$

I=3: normal distribution with mean *p1* and standard deviation *p2* ($p2 > 0$)

$$z = \Phi[(x - p1)/p2]$$

where Φ is the standard normal CDF and

$$f(x) = \frac{1}{\sqrt{2\pi} \;\; p2} exp[-\frac{(x - p1)^2}{2 \;\; p2^2}]$$

I=4: lognormal distribution with parameters *p1* and *p2* ($p1, \; p2 > 0$)

$$z = \Phi[(ln(x) - p1)/p2]$$

$$f(x) = \frac{1}{\sqrt{2\pi} \;\; p2} exp\{-\frac{[ln(x) - p1]^2}{2 \;\; p2^2}\}$$

VECTOR **VT** *T=tb,te* *N=n*

constructs an *n* vector containing a sequence of equally spaced data beginning at *tb* and ending at *te*. This command is used to generate the time or frequency coordinates for input functions.

$$\mathbf{VT}(i) = tb + (i-1) * (te - tb)/n$$

WRITE **M1** **FNAME** [*NR=nr1,nr2 NC=nc1,nc2*]

writes the contents of matrix **M1** into a new external file, named **FNAME**, where

M1 is an *n* x *m* matrix.

FNAME is a specified file name which does not exist in the working directory. The name should have less than six characters.

NR=nr1,nr2 are two row numbers. Only the data between these two rows are written in the external file. (Default: *nr1*=1, *nr2*=n).

NC=nc1,nc2 are two column numbers. Only the data between these two columns are written in the external file. (Default: *nc1*=1, *nc2*=*m*)