Time Series Modelling Version 4.32
Programming Reference

James Davidson

15th July 2010

Contents

Introduction

As well as running in GUI mode as a free-standing Windows or Linux application, TSM can be included as a module in a regular Ox program. This can be just an alternative way to run the program, using text commands. All the options available in the GUI version (with the exception of the graphics options, currently) can be implemented, by assigning values to TSM command variables. These are written in upper case, and are globally defined, and so can appear anywhere in the user's program.

It is also possible to run Ox code using the GUI version of TSM as a platform. See Appendix C of the main document for details of the functions that can be compiled and run as components of TSM. Communication between the program and the user's code is controlled through the dialog Model / Coded Function. In principle, any variable or function described in this document can be invoked from within a user's function.

Running a Program

Since TSM is a big program, a command line switch is needed to reserve more memory than the default. To run your program from OxEdit, first do the following.

1. Open OxEdit and choose View / Preferences / Add/Remove Modules…
2. Select the entry &Ox.
3. Edit the 'Arguments' field to read
   `-s6000,6000 "$(FilePath)"`
   (In other words, add the "`-s6000,6000`" switch at the beginning of the entry.)
4. Close the dialog. This setting will be remembered by the OxEdit installation.

The program needs to contain as its first line,

```
#import <packages/tsmod4/tsmknl4>
```

Otherwise it has the usual Ox structure, with a `main()` function where execution starts. A typical program would have the form

```
#import <packages/tsmod4/tsmknl4>
Text_Input()
{
     . . .
}
main()
{
     Set_Defaults();
     Text_Input();
     Run_Estimation();
}
```

where the ellipsis represents the options to be set. Each option must appear in a line having the form

```
   OPTION = [value];
```

where `OPTION` is one of a set of identifiers, and the user supplies `[value].` The terminating semi-colon is important. Note that Ox is case-sensitive, and the identifiers must be in upper case. The `main()` function must always appear last in the file, the

general rule being that *called* functions always precede *calling* functions. (To deviate from this rule, see the Ox documentation for more details.)
Comments in Ox (ignored by the compiler) are either placed between /*…*/ pairs, or are in lines beginning with //. These can be used for annotating the input file in any convenient manner.

Notes:

1. TSM is not an Ox class, just a precompiled module. This means that there are some globally defined variables whose use must be avoided in your program. All the user-selectable options are written wholly in upper case. This usage conflicts with the Ox convention of writing constants in upper case, but to avoid problems just don't use any word from the reserved list to define a constant! A complete alphabetized list of reserved words can be found in the file tsmknl4.h. Some other global definitions have the prefix g_ . A number of these are user-accessible and defined in this document. There are others, but they all relate to interaction with the GUI module. To avoid trouble, don't use the 'g_' prefix. Declare your global variables as STATIC, and use the prefix 's_'.
2. Since time is always short, the documentation of programming features tends to lag behind the development of the program itself. This manual is not always up to date. However, virtually all TSM features except graphics can be implemented in a user's Ox program. To see the commands needed to implement particular program features available in the GUI, give the command File / Settings / Display/Save Text… Please don't hesitate to advise the author of commands missing from this manual. For graphics, the use of either Ox graphics in OxMetrics, or GnuDraw graphics, is recommended.

Variable Types

1. Boolean: either `TRUE` (equivalently, 1) or `FALSE` (equivalently, 0).
2. Integer: whole numbers without decimal points.
3. Real: floating point numbers, can include decimal points.
4. String: alphanumeric characters enclosed in "".
5. Vector: values of types 1, 2 or 3, separated by commas, and enclosed by <>.
6. Matrix: values of types 1, 2 or 3 separated by commas and then semi-colons, and enclosed by <>. (For example, the 2×2 identity matrix is represented by <1,0;0,1>.)
7. Array: values of types 1-6, separated by commas and enclosed by {}.

Notes:

1. In Ox, row vectors are written with elements separated by commas, and column vectors with elements separated by semi-colons. All vector options in TSM are row vectors. Entering in column form will produce an error.
2. Vectors and matrices are used to input starting values for parameters. Except in the case of regime switching models, the entry takes the form of a single row.
3. If the vector or matrix you enter has fewer rows/columns than have been specified for estimation, it will be automatically extended with the default values. If it contains too many rows or columns it is truncated, and the additional ones are ignored.
4. In regime switching models with M = `NUM_REGIMES` regimes, matrices of switching parameters may have up to M rows, each row representing the starting values for a regime. If only a row vector is entered and `REGIME_DIFFERENCES` = 0, this row is automatically replicated M times to form the starting values. If

REGIME_DIFFERENCES = 1, then the additional rows are automatically set to zero.

5. Vectors and matrices are also used to input instructions about parameters, e.g. to fix them, or include them in a test of significance. In these cases the vector/matrix should contain ones and zeros, using the starting values as a template to identify the location of the parameter. The vector/matrix is extended with zeros/truncated, if the dimension is different from that specified.
6. The <> and {} symbols are optional if vectors/arrays have only one element.

TSM Functions

The following TSM functions can be called.

`Set_Defaults()`

Initializes program settings at default values. No return value.
Note: This function must always be called first, before any other TSM functions.

`Run_Estimation()`

Estimates currently specified model. No return value.

`Run_Simulation(const mShocks)`

Simulates currently specified model. No return value. The simulated series is optionally appended to the data matrix.
By default, set `mShocks` to the empty matrix <> . The random shocks are obtained from model residuals or through the random number generator, according to the options selected. Optionally, pass the shocks to the function as a matrix of dimension (`END_SAMPLE`–`START_SAMPLE+1`) × `columns(SERIES)`.
Note: A call to `Run_Estimation` must be normally made before a call to this function, to read in data and set up parameter values. Set `EVALUATE_INIT` = 1 to use supplied values instead of estimates.

`Load_TextValues()`

Loads parameter values and attributes that have been set manually using the formats described in Section 5. No return value.
Note: This function is called automatically when running programs in console mode. It *must* be called explicitly when running Ox code using the GUI version of TSM as a platform.

`SaveModel(const sTitle)`

Saves the current model specifications, including parameter values. `sTitle` is a text string containing a name used to identify the model in the output.This function returns an array containing the model, so the correct syntax is of the form "`aMod = SaveModel(sTitle)`".

`LoadModel(const aStoreModel, const iMode)`

Loads the model stored in the array `aStoreModel` in a previous call to `SaveModel`. This is equivalent to a call to a function `Text_Input()` containing the same specifications as TSM commands. Setting `iMode` = 1 loads all the values. Set `iMode` = 0 to avoid loading components that will not be used for simulations, including fixed values, upper and lower bounds, test values and testing options. These stay at their existing settings.

`ReadData(const sFile, const bSetSample, const bMerge, const iFirstnum)`

Reads a data set from a file.

`sFile` (string):
The path and name of the data file. The extension determines the type of file. (Remember that the Windows "\" symbol must be represented as "\\" in an Ox string variable.)

`bSetSample` (Boolean):

- 1, if all sample settings should be reset to the defaults (the complete sample),
- 0 ,otherwise.

`bMerge` (Boolean):

- 1, if the data are to be merged with the data set currently in memory
- 0, if the data are to replace the current data, if any.

`iFirstnum` (Integer):
When data sets are to be merged and the sample periods are different, set to the offset – the row number of the new data set that matches the first row of the existing data (can be of either sign). Otherwise set to 0.

Note:
This function needs to be called only if the data are to be manipulated in the user's program. Loading a model causes the associated data set to be loaded automatically. The data matrix and array of names are accessed through the static variables `DATA_SET` and `DATA_NAMES`, respectively. To have these data used in program functions, set `ACCESS_DATA` = 1.

`Summary_Statistics(const aSeries)`

Computes summary statistics and tests of I(0), I(1). The argument must be an array of variable names.

`LogPeriodogram_Regression(const bMode)`

Performs log-periodogram regression on elements of the array `LOGPER_SERIES`. Always set bMode = 1.

`Cointegration Analysis(const bMode)`

Performs a cointegration-related analysis, as follows.
`bMode` = 0: Perform tests of I(0) and I(1) on selected data.
`bMode` = 1: Prints selection criteria for lag length choice.
`bMode` = 2: Performs Johansen tests of cointegrating rank.
`bMode` = 3: MINIMAL analysis at 90% level.
`bMode` = 4: MINIMAL analysis at 95% level.
`bMode` = 5: MINIMAL analysis at 97.5% level.
`bMode` = 6: MINIMAL analysis at 99% level.
`bMode` = 7: Perform specified Wald test of cointegration.
`bMode` = 8: Perform all Walds test of cointegration.

`Run_MonteCarlo(const aSimMod, const aEstMod, const bExtendRun)`

Runs a Monte Carlo experiment. The first two arguments are the models to be used for generating the data and to be estimated, respectively. The third

argument is Boolean, indicating that the previous experiment is to be extended, instead of started afresh; set to 0 for most applications. No return value.
Notes: .

1. The models are named arrays, as previously created with `SaveModel`.
2. The simulation and estimation models can be the same or different, provided both reference the same data set. The observations for estimation are as specified in the simulation model.
3. The simulation model *must* explicitly assign the variables `START_SAMPLE` and `END_SAMPLE`. There are no valid default values.

`PrintCall(const bLine, …)`

If enabled, sends console output to a text file, (also the results window in GUI mode). `bLine` is a Boolean variable, = 1 to terminate the line with a carriage return, 0 otherwise. The other arguments are items for printing.
No return value.

In addition, the following functions can be used for accessing results inside the user's program after calls to `Run_Estimation()` or `Run_Simulation()`. See the section Accessing Results for details.

`LocP(const iEq, const iPar)`

This helps locate parameters and standard errors in multi-equation models.

`LocTP(const iReg, const iPar)`

This returns the storage locations of parameters and standard errors in Markov-switching models. See the section Accessing Results for details.

`LocVar(const Name)`

`Name` (string) is the name of a variable in `DATA_NAMES`.
Return value: column number of the variable in the matrix `DATA_SET`.
(This function has a different action from `VarNum`, which is only for use in user-supplied functions.)

In addition, the program can include `UserFunction()` and `UserSolve()` functions, as described in Appendix B. Don't forget to include the compiler directives `#define USER_FUNCTION` and `#define USER_SOLVE` in this case.
Note: Only certain listed TSM functions, as detailed in Appendix C, can be called from within user-supplied functions. Do not use the functions listed above in this context.

Options Reference

The numbering of these sections matches that in the User's Manual. The same program options are dealt with in the corresponding sections, as far as possible, and hence numbering is not consecutive. See the manual for some additional details.

***Tip:** The quickest way to learn the programming commands is to set up the desired model and options interactively in the TSM GUI, and then give the command File / Settings / Save Text. This command writes the contents of the `Text_Input()` function needed to generate the same run to a text file, ready for inclusion in your program.*

2.2 Data Input and Output

`ACCESS_DATA` (Boolean): Default = 0.

- 1, to access the data from a matrix created in the user's program
- 0, to read the data from a disk file.

`DATA_SET` (Matrix of Real)
The data matrix, with the observations in rows and variables in columns. Needs to be defined if `ACCESS_DATA` = 1.

`DATA_NAMES` (Array of Strings): Default = {}.
Names for variables, if these are either read from `DATA_SET` or from disk as an ASCII matrix file. Number of elements must match the number of columns of either `DATA_SET` or the ASCII file, respectively.

`INPUT_PATH` (String): Default = "".
Full path to the directory containing data files. The default (empty string) points to the working directory.

`INPUT_FILE` (String): Default = "input".
Name of the data file. Data can be read and saved in one of five formats. The format is specified by the file extension.

".xls" Excel Worksheet. The first row of the sheet must contain variable names.

".wk1", ".wks"
Lotus 123 worksheet. The first row of the sheet must contain variable names.

".in7" GiveWin file, see Ox and Givewin documentation for details.

".dat" Ox/Givewin "data with load information" file. See the documentation for details.

".mat", or any other.
ASCII file containing a matrix, with variables in columns and observations in rows. The first line of the file must contain two integers, number of rows, followed by number of columns.

EDF_FILE (String): Default = "".
Name of spreadsheet file containing empirical distribution function (see EDF_CRITS).
Note: See the GUI User's Manual, Section 3.10 (Setup / Monte Carlo Experiments) for details of the EDF file format.

ACCESS_RESULTS (Boolean) Default = 0.

- 0, write estimation results to console (normal output).
- 1, write estimation results to global variables, no console output.

This option is set when the estimation results are to be used in further processing by the program, as in Monte Carlo simulation. See the last section of this document.

RESULTS_FOLDER (String): Default = "".
Full path to the directory where results should be written, including text files and spreadsheet listings. The default (empty string) points to the working directory.

RUN_ID (Integer) Default = 0.
Initializes the numbering sequence used to identify outputs created by the program, including console output, listings files and retrieved series. It is incremented each time **Run_Estimation()** is called, and so lets the outputs of successive runs be easily distinguished.

3.1 Setup

DETERMINISTIC (Integer): Default = 0.

- 1 to remove mean and linear trend from the dependent variable by preliminary regression. This option is active only if the series are not differenced.
- 0 to remove the mean from the series prior to estimation. This option is active whether or not the series is differenced (see next setting).
- –1, no transformations applied to the series.

Note: This command is retained for legacy reasons, but estimating intercept and trend within the model is the recommended option.

START_SAMPLE (Integer): Default = 0.
First observation to be used for estimation. 0 is read as 1.

END_SAMPLE (Integer) Default = 0.
Last observation to be used for estimation. 0 is read as the last observation available.
Note: if START_SAMPLE > 1 then by default, the pre-sample observations are used to form lags. See PRESAMPLE_LAGS.

INDIC_SAMPLE (Boolean): Default = 0.

- 1, Select sample according values (1/0) of an indicator series in the data set. The series must have the name "!selectobs!".
- 0, otherwise.

OMIT_NANS (Boolean): Default = 0.

- 1, Omit missing observations from the selected sample without truncating. (Cross-section data only!)
- 0, otherwise.

3.5 Automatic Model Selection

MULTI_SPEC (Integer): Default = 0.

- 0, Normal Estimation

- 1, to estimate all the ARIMA(p,d,q) or ARFIMA(p,d,q) specifications in sequence, up to maximum values of p and q and $p+q$ set by the user. Starting values for each optimisation are generated automatically from the preceding run. If a conditional variance model is specified with `METHOD = 2`, the same specification and starting values, set by the user, are used for each specification of the mean process. Additional output options are not available in this case. Regime switching is disabled.
- 2, To compute models with all combinations of included regressors of the specified type(s), and report the case that optimizes the currently selected model selection criterion. (`INFO_CRIT`).

The following settings are ignored unless `MULTI_SPEC` = 1.

`MAX_AR_ORDER` (Integer): Default = 2.

Maximum value of p (order of $\phi(L)$) to be fitted.

`MAX_MA_ORDER` (Integer): Default = 2.

Maximum value of q (order of $\theta(L)$) to be fitted .

`MAX_TOTAL_ORDER` (Integer): Default = 2.

Maximum value of $p + q$.

Thus, with the default settings the program will estimate the following cases of (p,q), in the order shown: (0,0), (1,0), (2,0), (0,1), (1,1), (0,2). The starting values for the estimations are set to either the estimates from the preceding specification, or zero, as appropriate.

`AUTOREG_TYPES` (Integer): Default = 0.

Specified which regressor Types to include in the regressor selection run specified by `MULTI_SPEC` = 2.

1. Type 1.
2. Type 2
3. Type 3
4. Types 1 and 2
5. Types 1 and 3
6. Types 2 and 3
7. All Types.

3.5 Recursive/Rolling Estimation

`RECURSIVE_ESTIMATION` (Boolean): Default = 0.

- 1 to estimate the model repeatedly, for a sequence of samples with advancing end-dates.
- 0 regular model estimation.

The following commands are ignored unless `RECURSIVE_ESTIMATION` = 1. To use this feature, set `START_SAMPLE` and `END_SAMPLE` to represent the *first* sample in the desired sequence.

`ROLLING_ESTIMATION` (Boolean): Default = 0.

- 1 to estimate with fixed sample size, so that the start date and end date advance together,
- 0 to estimate with fixed start date, and increasing sample size.

`RECURSION_ENDDATE` (Integer): Default = 0.

The terminal end date in the sequence.

RECURSION_STEP (Integer): Default = 1.

The number of observations to advance by at each step.

RECURSION_STATISTICS (Boolean): Default = 1.

- 1 to save all the output, including test statistics, for each estimation
- 0 to save only parameter and estimates and standard errors.

FORECAST_TERMDATE (Boolean): Default = 0.

- 1 to compute ex-ante forecasts up to the fixed date, set as END_SAMPLE + FORECAST_STEPS, so that the actual number of forecasts steps contracts as the end-date of the samples advances.
- 0 to compute ex-ante forecasts a fixed number of steps ahead for each sample, so that the final forecast date advances with the sample.

Note: this option is ignored unless FORECAST_STEPS > 0 and EXPOST_FORECASTS = 0

SAVE_RECFORCS (Boolean): Default = 0.

- 1 to save all ex-ante forecasts in a file .
- 0 to save only the terminal forecast. This is recorded with the other model statistics.

RECURS_RPSTATUS (Boolean): Default = 0.

- 1, report convergence status of recursions .
- 0 otherwise.

DO_GRID (Boolean): Default = 0.

- 1, Compute a 1- or 2-dimensional grid of criterion values.
- 0 otherwise.

Note: the parameters to plot are set up in Fixed Values and Bounds: see User's Manual

3.6 Compute Summary Statistics

SUMMSTAT_DETREND (Boolean): Default = 0.

- 1, Compute statistics for detrended variables (LS residuals from trend).
- 0, Otherwise.

SUMMSTAT_DIFF (Boolean): Default = 0.

- 1, Compute statistics for differenced variables.
- 0, Otherwise.

SUMMSTAT_CORRELS (Integer): Default = 0.

Order of correlograms/partial correlograms to be calculated.

SUMMSTAT_DATCORR (Boolean): Default = 0.

- 1, *Either* compute contemporaneous data correlations for any number of series, *or* (if SUMMSTAT_CORRELS >0) compute cross-autocorrelations for a pair of series.
- 0, Compute summary statistics for individual series..

START_SSTSAMPLE (Integer): Default = 0.

First observation to be used for summary statistics. 0 is read as 1.

END_SSTSAMPLE (Integer) Default = 0.

Last observation to be used for summary statistics. 0 is read as the last observation available.

SUMMSTAT_INTORD (Integer): Default = 0.
Integration order tests to be performed.
- 0, No tests.
- 1, Tests of I(0).
- 2, Tests of I(1).
- 3, Tests of I(0) and I(1).

SUMMSTAT_QUANTILES (Boolean): Default = 0.
- 1, Compute quantiles of the series distribution.
- 0, Otherwise.

SUMMSTAT_PARCORREL (Boolean): Default = 0.
- 1, Compute partial correlograms.
- 0, Compute simple correlograms.

3.8 Semiparametric Long Memory

LOGPER_REGRESSION (Boolean): Default = 0.
- 1 Do log-periodogram regression.
- 0 Otherwise.

This is an indicator used by the Monte Carlo module.

LOGPER_SERIES (Array of Strings): Default = {}.
Variables for long memory estimation.
Note: The models are univariate. The named variables are estimated in sequence.

LOGPERIODGM_TRANS (Integer): Default = 0.
0, use raw series
1, use differenced series
2, use detrended series

LOGPERIODGM_TYPE (Integer): Default = 0.
- 0, Geweke/Porter-Hudak method.
- 1 Moulines-Soulier method.
- 2, Local Whittle ML

START_LPRSAMPLE (Integer): Default = 0.
First observation to be used for summary statistics. 0 is read as 1.

END_LPRSAMPLE (Integer) Default = 0.
Last observation to be used for summary statistics. 0 is read as the last observation available.

GPH_BANDWIDTH (Integer): Default = [(END_SAMPLE – START_SAMPLE)/2].
Bandwidth for Geweke/Porter-Hudak and local Whittle ML estimation.

GPH_SMOOTH (Integer): Default = 1.
Smoothing factor for Geweke/Porter-Hudak estimation.

GPH_TRIM (Integer): Default = 1.
Trimming factor for Geweke/Porter-Hudak estimation.

GPH_BIASBW (Integer): Default = [(END_SAMPLE – START_SAMPLE)/2]..
Bandwidth for Geweke/Porter-Hudak bias test.

MS_FOURTERMS (Integer): Default = 1.
Number of included Fourier terms for Moulines-Soulier estimation.

3.9 Cointegration Analysis

START_COISAMPLE (Integer): Default = 0.
First observation to be used for summary statistics. 0 is read as 1.

END_COISAMPLE (Integer) Default = 0.
Last observation to be used for summary statistics. 0 is read as the last observation available.

COINTEGRATION_VARS (Array of Strings): Default = `{}`.
Variables to include in the cointegrating VAR

COINTEGRATION_LAGS (Integer): Default = 1.
Lag length for cointegrating VAR.

COINTEGRATION_DRIFT (Boolean): Default = 0.
- 1, include trend term in cointegrating VAR.

0 otherwise.

COINTEGRATION_RANK (Integer): Default = 0.
Assumed cointegrating rank of system for MINIMAL analysis.

COINT_TEST_VARS (Array of Strings): Default = `{}`.
Subset of `COINTEGRATION_VARS` to include in Wald test of cointegration

MINIMAL_ROTHUMB (Boolean): Default = 0.
- 1, Use rule of thumb to adjust nominal rejection criteria in MINIMAL analysis.
- 0 otherwise.

3.10 Monte Carlo Experiment

MC_REPS (Integer): Default = 1000.
Number of Monte Carlo replications.

MC_BINS (Integer): Default = 100.
Number of bins for Monte Carlo empirical distributions.

MC_HISTOG (Boolean): Default = 0.
- 1, Report Monte Carlo distributions as a histogram.
- 0 otherwise.

Note: ignored unless `MC_REPS` > 1000.

MC_MOMENTS (Boolean): Default = 0.
- 1, Report first 4 empirical moments of parameters.
- 0 otherwise.

MC_MOMSES (Boolean): Default = 0.
- 1, Report first 4 empirical moments of parameter standard errors.
- 0 otherwise.

MC_CENTRET (Boolean): Default = 0.
- 1, tabulate distribution of centred *t* statistics
- 0 otherwise.

Note: ignored unless parameters match in generated and estimated models.

MC_COMPARE (Boolean): Default = 0.
- 1, compute bias and RMSE
- 0 otherwise.

Note: ignored unless parameters match in generated and estimated models.

MC_SIGNT (Boolean): Default = 0.

- 1, tabulate signed *t* statistics.
- 0 otherwise.

MC_QUANTILES (Boolean): Default = 0.

- 1, Tabulate EDFs for p-values .
- 0 otherwise.

MC_PEEVALS (Boolean): Default = 0.

- 1, Tabulate EDFs for p-values .
- 0 otherwise.

MC_ITGMM (Boolean): Default = 0.

- 1, Do iterated GMM in estimation.
- 0 Do 1-step GMM in estimation.

Note: ignored unless GMM is specified in estimation model.

4.1 Equation

METHOD (Integer): Default = `LSQ` .

- `LSQ` (0) for Least Squares.
- `WHITTLE` (1) for Whittle (frequency domain) ML.
- `GMM` (2) for Instrumental Variables / Generalized Method of Moments
- `GAUSS_ML` (3) for Conditional Gaussian time domain ML.
- `STUDENT_T` (4) for Conditional Student's *t* time domain ML
- `SKEW_STUDT` (5) Conditional skewed-Student's *t* time domain ML
- `GED_ML` (6) Conditional General error distribution time domain ML
- `PROBIT` (7) Probit ML (binary data).
- `LOGIT` (8) Logit ML (binary data)
- `POISSON` (9) Poisson ML (count data)
- `NEGBIN1` (10) Negative Binomial I (count data)
- `NEGBIN2` (11) Negative Binomial II (count data)

Notes:

1. *Either the variable name or the integer value can be given.*
2. *For the procedure for efficient (multi-stage) GMM, see 7.4 Optimization Options.*

SYSTEM (Boolean): Default = 0

- 1, System of equations.
- 0, single equation.

SERIES (String or Array of Strings): Default = "X".

The name(s) of the dependent variable(s) of the model.

Notes:

1. *If this option is given as an integer or row vector of integers, it is read as the relevant column number(s) of the data matrix.*
2. *With two or more variables, a system of equations is fitted.*
3. *A data series must be specified for a simulation run. This supplies start-up conditions (pre-sample lags), and acts as a 'placeholder' in the data matrix. As an alternative to a true data series, this command can supply the name of a dummy series composed (e.g.) of zeros.*

The following options are ignored if `METHOD = WHITTLE`*). In this case, the series is always de-meaned prior to computing the periodogram.*

`INTERCEPT_1` (Boolean): Default = 0

- 1 to fit an intercept of Type 1 .
- 0 to suppress intercept of Type 1.

`INTERCEPT_2` (Boolean): Default = 0

- 1 to fit an intercept of Type 2.
- 0 to suppress intercept of Type 2.

Note: if both intercepts are selected, the Type 2 selection will be ignored.

`TREND` (Boolean): Default = 0.

- 1 to fit a linear trend.
- 0, no trend

`LINEAR_REGRESSION` (Boolean): Default = 0.

- 1 to estimate an equation by ordinary (non-iterative) least squares or instrumental variables. In this case only the specifications in `REGRESSORS_1`, `REGRESSORS_2`, `TREND` and `INTERCEPT_1` are used.
- 0, normal operation

Note: Be careful to have `IS_ARFIMA`, `IS_GARCH`, `IS_FUNCTION`, *and* `IS_REGIMES` *set to 0.*

`ADF_TEST` (Boolean): Default = 0.

- 1 to compute the augmented Dickey Fuller cointegration test
- 0, otherwise

`PP_TEST` (Boolean): Default = 0.

- 1 to compute the Phillips-Perron cointegration test
- 0, otherwise

`FULLYMODIFIED_LS` (Boolean): Default = 0.

- 1 to compute Phillips-Hansen fully modified least squares estimates
- 0, otherwise

`SWSAIK_LS` (Boolean): Default = 0.

- 1 to compute Stock-Watson/Saikkonen augmented least squares estimates
- 0, otherwise

`IS_ARFIMA` (Boolean): Default = 1.

- 1, to enable ARMA/ARFIMA estimation
- 0, to disable ARMA/ARFIMA estimation; ignore all relevant settings in 4.4–5.5. Only conditional time domain ML estimation is available.

Provides a quick way to 'switch off' the time series options without changing all the lag settings.

`DIFFERENCING` (Boolean): Default = 0.

- 1, A unit root is imposed in estimation, equivalent to differencing the dependent variable(s) and regressors of Type 1.

- 0, otherwise.

Note: DIFFERENCING is ignored in linear regression, and when a user-coded function is specified.

AR_ORDER (Integer): Default = 0.

p, the order of $\phi(L)$ in equation (1).

MA_ORDER (Integer): Default = 0.

q, the order of $\theta(L)$ in equation (1).

Note: start/fixed/test/bound matrices have prefix ARMA_

NONLINEAR_MA (Boolean): Default = 0.

- 1 to implement the SPS nonlinear moving average model.
- 0 otherwise.

IS_DEE (Boolean): Default = 0.

- 1 to fit an ARFIMA(*p*,*d*,*q*) model.
- 0 to fit an ARIMA(*p*,1,*q*) or ARMA(p,q) depending on the setting of DIFFERENCING (see below).

Note: start/fixed/test/bound matrices have prefix DEE_

BILINEAR_ORDER (Integer): Default = 0.

r, the order of $\lambda(L)$ in equation (15).

REGRESSORS_1 (Array of Strings): Default = {}.
REGRESSORS_2 (Array of Strings): Default = {}.
REGRESSORS_3 (Array of Strings): Default = {}.

Note: start/fixed/test/bound matrices have prefixes REGR1_, REGR2_, REGR3_

These three options specify the vectors specified in equation2 (1) and (2). Each should supply an array containing the names of the variables in the data set to be included.

Notes:

1. *if $\theta(L) = 1$ in equation (1) then there is no distinction between x_{2t} and x_{3t}, and the contents of these vectors get the same treatment. Similarly for x_{1t} and x_{2t} if $d_1 = 0$ and $\phi(L) = 1$.*
2. *If the dependent variable is differenced, according to the DIFFERENCING option, then the regressors of Type 1 are also differenced automatically. Those of Types 2 and 3 are not.*
3. *If these options are vectors of integer, the entries are read as the relevant column numbers of the data matrix.*

REGR1_LAGS (Integer): Default = 0.
REGR2_LAGS (Integer): Default = 0.
REGR3_LAGS (Integer): Default = 0.

The number of lags of the variables of Types 1, 2 and 3 to be included in the model. If set to 0, only the current values are included.

Note: While all variables of the same Type must have the same order of lags, individual lag coefficients can be 'fixed' at zero; see Values.

4.2 Conditional Variance

IS_GARCH (Boolean): Default = 1.

- 1, to enable GARCHestimation.
- 0, to disable GARCH estimation; ignore all relevant settings in 4.6–4.8.

GARCH_AR_ORDER (Integer): Default = 0.

The order of $\delta(L)$ in equation (41) or equation (42) ("AR terms").

GARCH_MA_ORDER (Integer): Default = 0.

The order of $\beta(L)$ in equation (41) or equation (42) ("MA terms")

Note:

1. *Start/fixed/test/bound matrices have prefix GARCH_*
2. *the 'AR' and 'MA' terminology strictly applies only in equation (41). See the notes to GARCH_FORM for further information on the interpretion of the coefficients.*

IS_FGDEE (Boolean): Default = 0.

- 1 to estimate the FIGARCH or FIEGARCH models
- 0 for regular GARCH or EGARCH.

IS_HYGARCH (Boolean) : Default = 0. Ignored unless IS_FGDEE = 1.

- 0 for ordinary FIGARCH in (or FIEGARCH).
- 1 to estimate the HYGARCH model (see Davidson 2002).
- 2 for the HYGARCH variant defined in equation (12).

*Note: start/fixed/test/bound matrices have prefix FGDEE_*

APARCH (Boolean): Default = 0;

- 1 to estimate the APARCH model represented by equation (41) unrestricted.
- 0 to estimate the GARCH model in equation (2) with $\delta = 2$.

EGARCH (Boolean): Default = 0;

- 1 to estimate the EGARCH model represented by equation (3)
- 0 to estimate the GARCH model in equation (2).

Note:

1. *If APARCH is selected, this option is ignored.*
2. *there is a choice of algorithm for estimating EGARCH. See the ITERATE_EGARCH option.*
3. *The asymmetry parameter μ cannot be suppressed. Its value should be fixed at 0 to fit/test a symmetric version of EGARCH.*
4. *start/fixed/test/bound matrices for asymmetry parameter have prefix ASSYM_*

ASYMM_GARCH (Boolean): Default = 0;

- 1 to estimate the leverage parameter μ in equations (2) or (3).
- 0 otherwise.

Note:

1. *this option is ignored unless EGARCH = 0.*
2. *start/fixed/test/bound matrices have prefix ASSYM_*

DCC_GARCH (Boolean): Default = 0;

- 1 to estimate the DCC multivariate GARCH model, represented by equation (18) and (19).
- 0 Otherwise.

BEKK_GARCH (Boolean): Default = 0;

- 1 to estimate the BEKK multivariate GARCH model, represented by equation (20)
- 0 Otherwise.

Note: This option only applies to multiple equation models.

GARCH_REGRESSORS_1 (Array of Strings): Default = {}
GARCH_REGRESSORS_2 (Array of Strings): Default = {}.
GARCH_REGRESSORS_3 (Array of Strings): Default = {}.

These options specify the vectors of variables x_{4t}, x_{5t} and x_{6t}.

Notes:

1. *If these options are vectors of integer, the entries are read as the relevant column numbers of the data matrix.*
2. *start/fixed/test/bound matrices have prefixes* `GREG1_`, `GREG2_`, `GREG3_`

GREG1_LAGS (Integer): Default = 0.
GREG2_LAGS (Integer): Default = 0.
GREG3_LAGS (Integer): Default = 0.

The number of lags of the variables of Types 1, 2 and 3 to be included in the condtional variance model. If set to 0, only the current values are included.
Note: While all variables of the same Type must have the same order of lags, individual lag coefficients can be 'fixed' at zero; see Values.

GARCH_M (Boolean): Default = 0.

- 1 to include the conditional variance h_t as a regressor, in vectors x_{1t}, x_{2t} or x_{3t}, respectively.
- 0, otherwise.

GARCH_M_SD (Boolean): Default = 0.

- 1 to include the conditional standard deviation $h_t^{1/2}$ as a regressor, in vectors x_{1t}, x_{2t} or x_{3t}, respectively
- 0 otherwise.

Note: Only one GARCH_M regressor can be included. If both these options are set, `GARCH_M_SD` *is active and* `GARCH_M` *is ignored.*

GARCH_M_TYPE (Integer): Default = 1.

Type of GARCH-M regressor (1, 2 or 3)

4.4 User-coded Functions

(These options are ignored if `METHOD = WHITTLE`. See Appendix C for information on computing formulae as external Ox functions.)

IS_FUNCTION (Boolean): Default = 0;

- 1, to include a user-coded function $Y_t(\xi)$, in equation (1).
- 0, to include a measured series Y_t.

SUPPLIED_TEST (Boolean): Default = 0;

- 1, Compute a user-coded test statistic (see Appendix C) .
- 0, otherwise.

CODING_TYPE (Integer) Default = 0;

0. No coded equations.
1. Equation(s) are coded as strings in the array CODED_EQUATIONS. The formulae must have the form "[LHS variable] = [formula]".
2. Residual(s) or model components are coded as strings in the array CODED_EQUATIONS.
3. Coded nonlinear moving average function.
4. Coded nonlinear error correction mechanism.
5. Equation(s) coded in an external Ox function (returns residuals in same format as 2).
6. Log-likelihood terms coded as external Ox function.
7. Test statistic(s) for direct evaluation coded as external Ox function.
8. Data generated as external Ox function.

CODED_EQUATIONS (Array of Strings): Default = {};

These strings containing equation formulae must be defined if CODING_TYPE = 1 or 2. For details of the format see the GUI User's Manual, Sections 1.5 and 4.4. The number of array elements must be equal to the number of equations in the model (or number of equilibrium relations, see).

Notes:

1. *These commands is ignored unless IS_FUNCTION = 1. With this option, model specifications and estimation method are ignored. See Appendix C for details of implementing this option.*
2. *If IS_FUNCTION = 1, the value specified in SERIES is not used in computing the estimates. The dependent (normalised) variable is specified in the supplied code, if appropriate. However, the setting of SERIES will be used for headings in the output, and to select the data for actual and fitted values under PRINT_SERIES below. Be careful to set this option appropriately.*

FUNCTION_HEADING (String): Default = "".

An optional heading to appear in the output, identifying the model being fitted. Can also be used to identify the desired case in a library of user functions.

FUNCTION_NAMES (Array of Strings): Default = {};

Names for the parameters appearing in the user-supplied function, their order in the array corresponding to their positions in the vector.

Note:

1. *The number of elements in FUNCTION_NAMES is used by the program to indicate the number of parameters in the supplied function. It is the user's responsibility to ensure these correspond, otherwise a program crash will occur.*
2. *start/fixed/test/bound matrices have prefix FUNCTION_*

TEST_HEADING (String): Default = "".

An optional heading to appear in the output, identifying the test statistic being computed. Can also be used to identify the desired case in a library of user test statistics.

4.5 Regime Switching

(These options are ignored unless a maximum likelihood estimator is specified.)

IS_REGIMES (Boolean): Default = 0

- 1, Fit a switching regimes model.
- 0 otherwise. In this case, all subsequent settings in this section ignored.

NUM_REGIMES (Integer): Default = 1.
The number of regimes. Switching options are activated only if set to 2 or greater. The maximum allowed number of regimes is 4.

SWITCH_ITEMS (Vector of MEAN, VARIANCE, DEE, ARMA, REGR, VAR, STUDT, GARCH, FGDEE, GARCHREG, ASYMM, FUNCTION, EQUIL.): Default = <>.
This setting selects the parameter types that are to switch, using the usual identifiers. Any parameter types not listed will be held constant across regimes, and starting values entered as row vectors in the usual manner.
<MEAN> is equivalent to <DEE,ARMA,REGR,FUNCTION>
<VARIANCE> is equivalent to
<VAR,STUDT,GARCH,FGDEE,GARCHREG,ASYMM>
Example: SWITCH_ITEMS = <MEAN,VARIANCE> ; (i.e., all parameters switch.)
Notes:

1. *Only the entries MEAN and VARIANCE are active if the Hamilton model is selected.*
2. *To prevent a subset of parameters of a listed type from switching, select REGIME_DIFFERENCES and fix the differences at zero.*
3. *Integers can be entered in place of variable names. These must correspond to the variables' positions in the enumeration list, counting from 0, e.g. 0 for MEAN, 1 for VARIANCE, etc.*

HAMILTON_MODEL (Boolean): Default = 0.
- 1 to estimate the Hamilton/Hamilton-Susmel model of switch means and variances.
- 0 for simple Markor or explained switching.

HAMILTON_SWITCH (Vector of MEAN, VARIANCE): Default = <>.
Selects the parameter types to switch in Hamilton model, as in SWITCH_ITEMS. Should contain either or both of the entries MEAN and VARIANCE.
Note: Starting values for the mean and variance parameters must be entered in the first column positions of INT_START_VALUES and VAR_START_VALUES, respectively.

EXPLAINED_SWITCHING (Boolean): Default = 0.
- 1 to estimate a model with explained switching probabilities.
- 0 otherwise

Note: If EXPLAINED SWITCHING is selected, HAMILTON_MODEL is ignored.

EXPLSWITCH_REGIMES (Boolean): Default = 0.
- 1 to estimate a model with regime-dependent explained switching coefficients.
- 0 otherwise

SMOOTH_TRANSITION (Boolean): Default = 0.
- 1 to estimate a smooth transition model.
- **0** otherwise.

Note: if either EXPLAINED SWITCHING or HAMILTON_MODEL is selected, this setting is ignored.

SWITCHMOD_DUMS (matrix of Boolean): Default = < 0, 0, 0; 0, 0, 0; 0, 0, 0; 0, 0, 0 >

In the Explained Switching model, this matrix should be of maximum dimension 4×3. If final rows/columns are all zero, they can be omitted. Set the $(1, j)$ element to 1 to include an intercept in the equation for $\Pr(S_t = j \mid S_{t-1} = i)$. For $j = 2,\ldots,M$, set the (i, j) element to 1 to insert a shift dummy $I(S_{t-1} = i)$ in the equation for $\Pr(S_t = j \mid S_{t-1} = i)$.

In the Smooth Transition model, this matrix is of maximum dimension 1×2. Additional rows/columns are ignored. Set the (1,1) element to 1 to include an intercept in the single-transition model. Set the (1,2) element to 1 to include an intercept in the double-transition case.

SWITCH_REGRESSORS_1 (Array of Strings): Default = {}

SWITCH_REGRESSORS_2 (Array of Strings): Default = {}.

SWITCH_REGRESSORS_3 (Array of Strings): Default = {}.

These options specify the vectors of explanatory variables to appear in the function $\Pr(S_t = j \mid S_{t-1} = i)$ for $j = 1,\ldots, M - 1$.

Note:

1. *Each specification defines a column of the transition matrix. The rows optionally differ by shift dummies.*
2. *start/fixed/test/bound matrices have prefix* `SWREGR__`

REGIME_DIFFERENCES (Boolean): Default = 0.

- 1 to estimate parameters for Regimes $2,\ldots,M$ as differences from Regime 1
- 0 to estimate the actual parameters for each regime.

SWITCH_LAG (Integer): Default = 0;

Lag of switch regressors.

4.6 Parameter Constraints

IS_CONSTRAINTS (Boolean): Default = 0.

- 1, set up parameter constraints.
- 0, otherwise

WALD_TEST (Boolean): Default = 0.

- 1, use parameter constraints to compute a Wald test.
- 0, impose parameter constraints in estimation

RESTRIC_TYPE (Integer): Default = 0.

- 0, set up multiple zero (exclusion) restrictions.
- 1, set up r linear restrictions of the form $R\theta = c$, where θ $(n \times 1)$ is the full vector of parameters, R $(r \times n)$ is a matrix of fixed coefficients and c $(r \times 1)$ is a vector of constants.
- 2, read Ox-coded parameter restriction(s) from file (see Appendix C)

TEST_CONSTANTS (Vector/Matrix of Reals): Default = <0>. Ignored unless `RESTRIC_TYPE` = 1.

A vector containing the elements of c (transposed).

ALLREGRS_TEST (Boolean): Default = 0

- 1, Compute a Wald test of all included regressors, excluding lagged dependent variables, intercept and trend.
- 0, otherwise.

RESTRIC_TEXT (String) : default = "";

Text of the Ox-coded restriction(s). Saved to file `user_restrictn.txt` when a model is saved . Also appears in the output.

4.7 Equilibrium Relations

IS_ECM (Boolean): Default = 1.

- 1, the equation(s) of the system contain one or more error correction terms.
- 0, otherwise.

ECM_TERMS (Integer): Default = 0.

The number of equilibrium relations to be included.

ECM_LAG (Integer): Default = 1.

The lag to be assigned to the equilibrium relations.

EQUIL_VARIABLES (Array of Strings): Default = {}.

Variables to include in the equilibrium relations.
Note: restrictions are imposed using EQUIL_FIXED_VALUES.

VECM_TYPE (Integer): Default = 0.

0. variables selected in `EQUIL_VARIABLES`
1. equililibrium relations automatically contain dependent variables *less* fitted intercepts and Type 1 regressors.
2. coded equilibrium relations. The formula(e) must be entered in `CODED_EQUATIONS`.

Note: This command is ignored unless `IS_DEE` = 1 and `IS_ECM` = 1.

NLECM_TYPE (Integer): Default = 0.

- 0, linear ECM
- 1, nonlinear ECM exponential smooth transition
- 2, Asymmetric ECM
- 3, Cubic ECM

FRAC_ECM (Boolean): Default = 0.

- 1, fractional cointegration is implemented – error correction terms are fractionally differenced.
- 0, otherwise.

Note: This command is ignored unless `IS_DEE` = 1 and `IS_ECM` = 1.

COMMON_FRAC (Boolean): Default = 0.

- 1, in a system of equations, the fractional integration parameters for Equations 2,… are estimated as differences from the same parameters in Equation 1.
- 0, otherwise.

Note: This command allows equality of the parameters across equation to be easily imposed and tested. It is ignored in single equation models. However, it applies to all fractional parameters, whther or not fractional cointegration is specified.

GENERALIZED_COINT (Boolean): Default = 0.

- 1, generalized fractional cointegration is implemented – components of cointegrating vectors are fractionally differenced.
- 0, regular cointegration.

Note: This command is ignored unless `IS_DEE` = 1 and `IS_ECM` = 1.

4.8 Select Instruments

INSTRUMENTS (Array of Strings): Default = {}. Instruments for GMM estimation.

INSTR_INTERCEPT (Boolean): Default = 1.

- 1, include intercept in the instrument set.
- 0, otherwise

INSTR_TREND (Boolean): Default = 0.
- 1, include trend in the instrument set.
- 0, otherwise

LAG_INSTRUMENTS (Boolean): Default = 0.
- 1, use lagged endogenous variables as instruments.
- 0, otherwise.

INSTR_LAGS (Integer): Default = 0.

Number of lags of additional instruments to be included.

4.9 Panel Data

Panel data estimation requires the data file to be formatted in a specified manner; see the User's Manual Section 2.2 for details. When the data are read in this format, the following commands can be set. Please note that many other estimation and testing options are unavailable for panels. These options in general will do nothing, if set, but could conceivably cause a program crash. If in doubt, make sure that doubtful options are deleted from the input file.

PANEL_TRANSFORM (Integer): Default = 0.
- 0, No transformation
- 1, Individual mean-deviations
- 2, Individual means
- 3, time first-differences
- 4, orthogonal deviations

PANEL_TDUMS (Boolean): Default = 0.
- 1, Include time dummies
- 0, Otherwise

PANEL_INDVDUMS (Boolean): Default = 0.
- 1, Include individual dummies
- 0, Otherwise

PANEL_GPDUMS (Boolean): Default = 0.
- 1, Include group dummies
- 0, Otherwise.

PANEL_MTHD (Integer): Default = 0.
- 0, OLS (fixed effects)
- 1, Feasible GLS (random effects)
- 2, Maximum likelihood (random effects)

The following two settings are used to generate Gaussian shocks for simulations. They are <u>not</u> estimation inputs. However, note that the estimated values of these parameters are written to these locations after an estimation run.

PANEL_SIGV (Real): Default = 0. Variance of within-individual disturbances.

PANEL_TAU (Real): Default = 0. Ratio of between-individual to within-individual variances.

5. Values

Values are entered in matrices having identifiers with the general format
[parameter group]_START_VALUES (real)
[parameter group]_FIXED_VALUES (Boolean)
[parameter group]_UPPER_BOUND (real)
[parameter group]_LOWER_BOUND (real)
[parameter group]_TEST_VALUES (array: first element Boolean, others (optional) real),
where [parameter group] is one of **DEE, AR, MA**, **BAR, BMA**, **NMA**, **INT**, **REGR1**, **REGR2, REGR3, VAR**, **GAR**, **GMA**, **FGDEE**, **GREG1, GREG2**, **GREG3, ASYMM**, **FUNCTION**, **STUDT**, **MARKOV**, **SWREGR1**, **SWREGR2**, **SWREGR3, EQUIL, CORREL, FRACTPI.**

If an element of the [parameter group]_FIXED_VALUES matrices is set to 1 (or TRUE), the corresponding parameter is fixed at its starting value or, in the case of a grid evaluation, at grid values.

If an element of the first array element of [parameter group]_TEST_VALUES is set to 1 (or TRUE), the corresponding parameter is included in the Wald test restriction. The second and subsequent elements, which should be present when RESTRIC_TYPE = 1 and otherwise are ignored, define linear restrictions on the parameters in conjunction with the vector TEST_CONSTANTS.

The columns of these matrices correspond to parameters, and the rows to different regimes, where the parameters in question are switching. In non-switching models, or where no switching is specified for the group, they are simply row vectors. The [parameter group]_START_VALUES matrix can be regarded as a template for the others, their elements acting on the parameters in corresponding positions.

Systems of Equations

To specify a system, an array of matrices is required (one matrix for each equation) for the following groups: DEE, AR, MA, BAR, BMA, NMA, INT, REGR1, REGR2, REGR3, VAR, GAR, GMA, FGDEE, GREG1, GREG2, GREG3, ASYMM, FUNCTION. The usual matrices only are required for STUDT, MARKOV, SWREGR1, SWREGR2, SWREGR3, CORREL, FRACTPI. An array is also required for EQUIL, in this case, one for each equilibrium relation.

Remember that each equation in a system has the same nominal specification, so each matrix of a group has the same number of columns. Restrictions are imposed by the settings of [parameter group]_FIXED_VALUES to specify differences between the equations.
The items [parameter group]_TEST_VALUES must now be constructed an array of arrays. The elements of the outer array correspond to the equations, each of them having tests specified by the inner arrays, as for the single equation case.

The Parameter Groups

If the matrices entered do not match the dimensions indicated, they are padded with default values, or trimmed, as appropriate. Therefore, a matrix only has to be entered if its contents differs from the defaults.

DEE : rows = 1 or *M*, columns = 1. ARFIMA *d*.

Note: When `MULTI_SPEC = 1` *and, the Robinson (1994) nonparametric estimate of d is used as a starting value for the initial model fitted, the ARFIMA(0,d,0). Thereafter, the currently estimated value is used.*

AR: rows = 1 or *M*, columns = `AR_ORDER.` AR coefficients.

MA: rows = 1 or *M*, `MA_ORDER`, default = <>. MA coefficients.

BAR: rows = 1 or *M*, columns = `AR_ORDER`. Bilinear AR coefficients.

BMA: rows = 1 or *M*, columns = `BILINEAR_ORDER` – 1. Bilinear MA coefficients.

NMA: rows = 1 or *M*, columns = 5. Nonlinear MA parameters, in the order α, β, γ, c_1, c_2.

INT: rows = 1 or M, columns = 1. The intercept.

Note: This is the value of the intercept whether of Type 1 or Type 2.

REGR1, REGR2, REGR3 : rows = 1 or *M*, columns = number of regressors specified of Types 1, 2 and 3, including trend (must be of type 1) and GARCH-M term. The trend should be listed *after* any observed variables, The GARCH-M term should be *last* in its assigned type.

Notes:

1. *When deleting or adding regressors, don't overlook that the starting values must be edited to match, or else the wrong values will get used, without prompting.*
2. *When lags are specified using* `REGR1_LAGS`*, etc., in general the number of columns equals the number of variables times* 1+ *the number of lags. The exception to this rule is in* `REGR2` *, where if a dependent variable (listed in* `SERIES`*) is also listed in* `REGRESSORS_2` *, the current value is omitted (count lags from 1, not from 0).*
3. *If, in system estimation, dependent variable(s) are listed in* `REGR1`*, the system is treated as simultaneous, and the FIML estimator is implemented. The "own" dependent variable will have its coefficient fixed at zero automatically, if this is not done manually.*
4. *If equilibrium relations are specified, the ECM coefficients are located in* `REGR2`*, after the other variables.*

GAR: rows = 1 or *M*, columns = `GARCH_AR_ORDER` GAR coefficients

GMA: rows = 1 or *M*, columns = `GARCH_MA_ORDER`, GMA coefficients

Note: Make sure starting values match the `GARCH_FORM` *and* `MA_FORM` *settings. If, in equation (41) , the signs of the starting values do not observe GARCH positivity constraints, they are reversed. If starting values are either not set, or not defining fixed values, they are set to 0.1.*

FGDEE: rows = 1 or *M*, columns = 2, FIGARCH *d* followed by HYGARCH amplitude parameter α.

Note: unless starting values satisfy $0 < d \leq 1$ *and* $\alpha > 0$ *(and are not fixed values) each is reset to 1.*

GREG1, **GREG2**, **GREG3**: rows = 1 or *M*, columns = number of GARCH regressors of each Type specified, similar to `REGR1`. etc.

ASYMM: rows = 1 or *M*, columns = 1, TARCH asymmetry or EGARCH asymmetry parameter.

VAR: rows = 1 or *M*, columns = 1. Error variance,.

FUNCTION: rows = 1 or *M*, columns = number of names specified in supplied function.

STUDT : rows = 1 or *M*, columns = 2.

Parameter(s) depend on the likelihood function selected:

- For Student's *t* distribution, degrees of freedom parameter and skewness parameter if specified.
- For the GED distribution, ν parameter (second column empty).
- For Negative Binomial I and II, α parameter (second column empty).

MARKOV: rows = *M*, columns = *M* –1. These are the fixed Markov transition probabilities, $\Pr(S_t = j|S_{t-1} = i)$, with $i = 1,\ldots,M$ corresponding to rows and $j = 1,\ldots,M-1$ to columns – the *M*th column is not entered, and is defined by the identity that the rows sum to unity

Notes:

1. *If the starting values sum to more than 1, they are ignored and the default values 1/M are used.*
2. *For estimation, the probabilities are mapped to the real line by a logistic transformation.*

SWREGR1, SWREGR2, SWREGR3: rows = 1, columns = number of explained switching variables specified in regimes 1, 2 and 3. These must appear with the intercepts first in the list, followed by regime dummies, followed by variables.

A maximum of 4 regimes is allowed, and hence at most three independent models determine the probabilities of switching to a regime.

EQUIL : rows = 1 or *M*, columns = number variables specified in `EQUIL_VARIABLES`.

Note: At least one coefficient in each equilibrium relation must be fixed. The program will fix the first element of each relation to 1, automatically, if none is fixed manually.

CORREL : rows = 1 or *M*, columns = $N(N-1)/2$ where *N* = number of equations.

Note: These coefficients represent the correlations of the equation errors. Since the latter are constrained to lie in (-1,1), they are defined as $C/(1+|C|)$ *where C is the value set here.*

FRACTPI : rows = 1 or *M*, columns = number of included pre-sample proxy terms (recommended values, 1 or 2).

Inequality Constraints

IS_BOUNDS (Boolean): Default = 0.

- 1, to enable inequality constraints
- 0, to disable inequality constraints

[parameter type]_LOWER_BOUND (Vector/Matrix of Real) Default = <>,
[parameter type]_UPPER_BOUND (Vector of Real) Default = <>..

These vectors are formatted just like the corresponding [parameter type]_START_VALUES vectors.

Note:

1. *The bounds are ignored unless* `IS_BOUNDS` *is set to 1, and the upper one strictly exceeds the lower. Cancel the setting by putting both bounds to 0.*
2. *It is not recommended to use this technique routinely to impose, e.g., positivity and stability constraints on lag polynomials. The search algorithm should usually work OK without this. The method is implemented more as a last resort for difficult cases.*

Constraint Values

There are two cases of **[parameter group]_TEST_VALUES**, selected with RESTRIC_TYPE.

1. *Zero Restrictions.* The arrays have one element, a Boolean matrix, with elements set to 1 (or TRUE) for each parameter to be constrained to zero, and 0 (or FALSE) otherwise.
2. *Linear Restrictions.* The arrays have $r + 1$ elements where r = the number of linear restrictions. The first elements are Boolean matrices, to indicate inclusion of the parameter in the restrictions. The remaining elements contains the relevant segment of the row of R.

Important note:

When running Ox code using the GUI version of TSM as a platform, call the function Load_TextValues() immediately after setting parameter attributes.

6. Actions

PRINT_INFO ((Boolean): Default = 0.

- 1 to call the Ox Database function Info, giving descriptive statistics on the data set.
- 0 to suppress this output.

PRINT_SUMMSTATS (Boolean): Default = 0.

- 1 to print summary statistics and tests of the order of integration for the dependent variable.
- 0 to suppress this output.

The following statistics are printed

1. Lo's (1991) R/S test for I(0) against I(d) for $d > 0$ or $d < 0$.
2. The Kwiatkowski et al. (1992) (KPSS) test of I(0), against I(1).
3. Phillips-Perron (1988) test on the *cumulated* series, hence of I(0) against I(−1).
4. The Robinson (1994) semiparametric estimator of d.

SUMMSTAT_CORRELS (Boolean): Default = 0, ignored unless PRINT_SUMMSTATS = 1.

- 1 to print autocorrelations and Q statistics for levels and squares of the dependent variable.
- 0 to suppress this output.

EVALUATE_INIT (Boolean): Default = 0.

- 1, to print the specified output at the input parameter starting values.
- 0, for normal optimisation procedure.

Use this option to print listings and test results without repeating a lengthy optimisation.

DO_GRID (Boolean): Default = 0.

- 1, to compute a grid of criterion values at fixed, equally spaced values of one or two parameters, while optimising over the remaining parameters.
- 0, for normal optimisation procedure.

Use this option to create a contour plot of the concentrated criterion function. The operation is carried out on the first one or two parameters satisfying the following conditions: 1) the "fixed" flag is set; 2) the upper bound exceeds the lower bound. Also see Inequality Constraints and `GRID_POINTS`.

`MULTISTAGE_GMM` (Boolean): Default = 0.

- 1 to use the efficient GMM minimand, after evaluating the residuals at the starting values of the parameters.
- 0 to use the standard instrumental variables minimand.

Notes:

1. *To do multi-stage GMM, call `Run_Estimation()` two or more times in succession, with `MULTISTAGE_GMM` = 0 for the first run and `MULTISTAGE_GMM` = 1 for the subsequent runs. The starting values for these runs will be the convergence point of the previous run.*
2. *This setting is ignored if `GRID_PLOT` = 1 or `MULTI_SPEC` = 1 or `LINEAR_REGRESSION` = 1 .*

`SPEC_FORCS` (Boolean): Default = 0.

- 1, compute currently specified forecasts at current parameter values.
- 0, otherwise

`SPEC_DIAGS` (Boolean): Default = 0.

- 1, compute currently specified diagnostic tests at current parameter values.
- 0, otherwise

`SPEC_MTEST` (Boolean): Default = 0.

- 1, compute currently specified M-test at current parameter values.
- 0, otherwise

`SPEC_SCRTEST` (Boolean): Default = 0.

- 1, compute currently specified score test at current parameter values.
- 0, otherwise

`SPEC_WALD` (Boolean): Default = 0.

- 1, compute currently specified Wald test at current parameter values.
- 0, otherwise

`LMTEST_TYPE` (Integer): Default = 0.
Determines the location of the test variables in a specified score test.

- 0, Conditional Mean model , Type 1 regressor(s)
- 1, Conditional Mean model, Type 2 regressor(s)
- 2, Conditional Mean model, Type 3 regressor(s)
- 3, Conditional Variance model, Type 1 regressor(s)
- 4, Conditional Variance model, Type 2 regressor(s)
- 5, Conditional Variance model, Type 3 regressor(s)

Note: If the selected regressors Type is specified to include lags, the test variables are lagged similarly. The degrees of freedom of the test are then "number of variables × 1 + number of lags".

`SCRTEST_VARIABLES` (Array of strings): Default = {}

Names of variables from the data set to use as indicator variables in the specified score (LM) test . Specification similar to REGRESSORS_1 etc.

CONDITIONAL_MTEST

- 1, to compute a conditional moment test. (Variance matrix computed by outer-product formula.)
- 0, to compute a simple moment test (variance matrix computed by HAC formula.)

SQUARES_MTEST (Boolean): Default = 0.

- 1, to use the squared normalized residuals as the M-test or CM-test covariate.
- 0, to use the normalized residuals as the M-test or CM-test covariate.

MTEST_VARIABLES (Array of strings): Default = {}

Names of variables from the data set to use as indicator variables in the specified M-test . Specification similar to REGRESSORS_1 etc.

RECYCLE_RESULTS (Boolean): Default = 0.

- 0, no recycling.
- 1, to print the parameter estimates in "input-ready" form.

This option creates formatted lines ready to be cut and pasted into the run file. This provides a simple way to repeat the previous run using some or all of the estimates as starting values. Used in combination with EVALUATE_INIT, it allows tests and listings to be obtained for a previous run without repeating the optimisation procedure. *Note: If ACCESS_RESULTS = 1, then with RECYCLE_RESULTS = 1 the estimates are written internally as starting values for the next run. This is appropriate if the user's program specifies a succession of estimation runs, where the next run should be started as near as possible to the estimates obtained on the previous run.*

8.2 Output and Retrieval Options

RETRIEVE_RESIDUALS (Boolean): Default = 0.
RETRIEVE_FITTED (Boolean): Default = 0.
RETRIEVE_VARADJRES (Boolean): Default = 0.
RETRIEVE_CONDVARS (Boolean): Default = 0.
RETRIEVE_PROBS (Boolean): Default = 0.
RETRIEVE_SIM (Boolean): Default = 0.
RETRIEVE_EQUILS (Boolean): Default = 0.

- 1, to retrieve the created series specified, and append it to the data set.
- 0, otherwise.

The effect of these commands depends on the model specified.
RETRIEVE_VARADJRES (variance-adjusted residuals) and RETRIEVE_CONDVAR (conditional variances) are active only when GARCH-type and/or regime switching models with switching variance components are specified.
RETRIEVE_PROBS (Markov filter probabilities, explained switching probabilities, and regime weights in smooth transition models) is active only in regime-switching models.
RETRIEVE_SIM retrieves the results of a call to Run_Simulation(0), otherwise it has no effect.

RETRIEVE_EQUILS retrieves equilibrium relations when an ECM model is specified, otherwise it has no effect.

SHOW_CRITERIA (Boolean): Default = 0.

- 1 to print the log-likelihood and model selection criteria
- 0 otherwise.

PRINT_COVMATRIX (Boolean): Default = 0.

- 1 to print the covariance matrix of the parameters
- 0 otherwise.

PRINT_CORRELS (Boolean): Default = 1.

- 1 to print the correlograms of the residuals
- 0 otherwise.

PRINT_SERIES (Boolean): Default = 0.

- 1 to send series (residuals etc.) to the console.
- 0 otherwise.

PRINT_LISTINGS (Boolean): Default = 0.

- 1 to send forecasts and impulse-response coefficients to the console.
- 0 otherwise.

PRINT_RESULTS (Boolean): Default = 1.

- 1 to send results to the console.
- 0 otherwise.

OUTPUT_RESULTS (Boolean): Default = 0.

- 1 to append results to a text file.
- 0 otherwise.

AUTOSAVE_LISTS (Boolean): Default = 0.

- 1 to save listings (residuals, forecasts etc.) to a file, of type specified by the setting of OUTPUT_SERIES.
- 0 Listings not saved to file

OUTPUT_SERIES (Integer): Default = 0.

- 0 save listings to Excel spreadsheet (.XLS)
- 1 save listings to Lotus 123 spreadsheet (.WKS)
- 2 save listings to GiveWin file (.IN7)
- 3 save listings to data file with format information (.DAT)
- 4 save listings to matrix file (.MAT)

8.3 Test and Diagnostics Options

Q_TEST (Integer): Default = 0.

- 0, no *Q* autocorrelation test,
- 1 to use the Box-Pierce (1970) formula for the *Q* autocorrelation test,
- 2 to use the Ljung-Box (1978) formula for the *Q* autocorrelation test.

Q_TEST_ORDER (Integer): Default = 12.

The numbers of lags to be used in computing the Box-Pierce (1979) and McLeod-Li (1980) diagnostic statistics. (Also see LJUNG_BOX).

CORRELOGRAM_ORDER (Integer): Default = 0.

The number of residual correlogram points of residuals and squared residuals to be reported, if any. Set to 0 for no correlogram. This choice is independent of the Box-Pierce order, note.

COVMAT_TYPE (Integer): Default = 0.

- 0 standard covariance matrix formula
- 1 robust covariance matrix formula.
- 2 HAC covariance matrtix formula

LM_TEST (Boolean): Default = 0.

- 1 to print the LM statistic for restrictions imposed with the "fix parameter" flags (see VALUES).
- 0 otherwise

SCORE_TEST (Boolean): Default = 0.

- 1 To compute diagnostic score (LM) statistics, as specified by DIAGNOSTIC_TESTS.
- 0 otherwise

MOMENT_TEST (Boolean): Default = 0.

- 1 To compute diagnostic conditional moment (CM) statistics, as specified by DIAGNOSTIC_TESTS.
- 0 otherwise

DIAGNOSTIC_TESTS (1 × 16 vector of Boolean): Default = zeros(1,16).

Set one or more elements to 1 to compute a diagnostic test statistic. The cases (counting columns from 0) are

[0] = Autocorrelation;
[1]= Neglected ARCH;
[2] = Nonlinear Functional Form;
[3] = Heteroscedasticity;
[4] = AR Common Factor (COMFAC) test
[5] = Information Matrix test.
[6] = Durbin-Watson test
[7] = KPSS test
[8] = Lo's R/S test
[9] = HML long memory test (also see HML_SETTINGS)
[10] = Nyblom-Hansen specification test
[11] = Individual Nyblom-Hansen tests on score elements.
[12] = Bootstrap test of I(0) hypothesis.
[13] = Bierens' specification test
[14] = User-coded test
[15] = White's heteroscedasticity test.

The type of tests computed in cases [0]-[3] depend on the settings of SCORE_TEST and MOMENT_TEST. Either test , or both tests, can be specified. The information matrix test is computed regardless of these settings

HMLTEST_C (Real): Default = 1.
Truncation parameter (*c*) for HML test.

HMLTEST_L (Real): Default = 2/3
Bandwidth parameter (*L*) for HML test.

BIERENS_BOUND (Real): Default = 1/2
Bounding constant (γ) for modified Bierens test.

BIERENS_POWER (Real): Default = 1/2
Power of sample size (ρ) for modified Bierens test.

DGTEST_LAGS (Integer): Default = 1.
Number of lags for diagnostic tests of autocorrelation.

DGTEST_SQLAGS (Integer): Default = 1.
Number of lags for diagnostic tests of neglected ARCH.

DGTEST_FFORDER (Integer): Default = 2.
Maximum power of fitted values to include in test of functional form.

NOPRINT_OUTPUT (Boolean): Default = 0.

- 1 To suppress printed output except for test results. Use this setting in conjunction with `EVALUATE_INIT` to compute tests from the current estimates, without re-estimating.
- 0 otherwise.

COVMAT_TYPE (Integer): Default = 1;

- 0, Standard covariance matrix formula (information matrix)
- 1, Robust - heteroscedasticity consistent formula, also valid for quasi-likelihood applications.
- *3.* HAC – heteroscedasticity and autocorrelation consistent ('Newey-West') formula

Note: if specified, robust and HAC formulae are used to compute the standard errors/covariance matrix/Wald and LM tests and also preliminary tests of I(0) (KPSS, Phillips-Perron, Lo R/S).

KERNEL_TYPE (Integer): Default = 1;

- 0, No kernel (heteroscedasticity correction only, equivalent to `COVMAT_TYPE` = 1);
- 1, Parzen kernel
- 2, Bartlett kernel
- 3, Quadratic Spectral kernel
- 4. Tukey-Hanning kernel.

HAC_BANDWIDTH (Integer): Default = 0;
If this setting is positive, its value is used for the bandwidth. If it is zero, the program defaults are used; $[n^{1/3}]$ for the Bartlett kernel, and $[n^{1/5}]$ for the other cases.

INFO_CRIT (Integer): Default = 1;
Criterion for lag choice in for ADF tests and S-W/Saikkonen estimates

- 0, Akaike criterion
- 1, Schwarz criterion

- 2, Hannan-Quinn criterion.

EDF_CRITS (Boolean): Default = 0;

- 1, Use supplied empirical distribution to provide test critical values
- 0, Otherwise.

Note: The EDF is read from disk as a spreadsheet file; see **EDF_FILE**.

8.4 Forecasting Options

FORECAST_STEPS (Integer): Default = 0.

The number of multi-step forecasts of Y_t to be computed. This option generates three series: the point forecasts, and ±2 standard error bounds. If a GARCH model is fitted, the bounds are computed using the m-step ahead conditional variance forecasts.

MOVING_AVERAGE_COEFFS (Integer): Default = 0.

The number of solved moving average lag coefficients from the mean process and (where fitted) the variance process, to be listed.

EXPOST_FORECASTS (Boolean): Default = 0

- 1, To compute one-step ex-post forecasts (using actual values of lags)
- 0, To compute dynamic ex-ante multi-step forecasts.

MONTECARLO_FORECASTS (Boolean): Default = 0

- 1, To compute ex-ante forecasts by Monte Carlo stochastic simulation.
- 0, Yo compute ex ante forecasts by analytic formulae for mean and variance.

MCFORECAST_TYPE (Boolean): Default = 1

- 1, To report medians of Monte Carlo distributions with 95% confidence bands.
- 0, To report means of Monte Carlo distributions with 2SE bands.

MCFORECAST_REPLICATIONS (Integer): Default = 1000.

Number of replications in Monte Carlo forecasts .

FCST_SEBANDS

- 1, To compute forecast confidence bands.
- 0, Otherwise.

Notes:

1. *The analytic forecast standard errors are asymptotic, and do not take account of parameter uncertainty.*
2. *For ex-post forecasting, or if the model contains regressors from the data set, the number of forecasts is limited to the available post-sample observations. However, the trend dummy and the GARCH_M regressor are extended as necessary.*
3. *Analytic standard error bands cannot take account of dynamics introduced through a bilinear or user-supplied function. Use Monte Carlo forecasting in these cases.*
4. *Unless estimation is by maximum likelihood, only Gaussian or bootstrap random numbers are available.*

8.5 Simulation and Resampling Options

SIM_DISTRIBUTION (Integer): Default = 1.
The method of generating shocks for stochastic simulation of the current model, for use in one-off simulations, Monte Carlo forecasts and bootstrap tests.

- 1, to generate random numbers from the distribution specified by the likelihood function - Gaussian, Student's *t* or skewed Student, using current parameter values).
- 2 , to generate Gaussian random numbers, with zero mean and variance either set by the user, or that of the current residuals.
- 3, to use the bootstrap or block bootstrap, resampling the current residuals. They are centered and bias-adjusted by $n/(n-k)$. Not available unless a model has been estimated.
- 4, to generate random numbers using the wild bootstrap. Not available unless a model has been estimated.
- 5, to generate random numbers using the sieve-AR bootstrap. Not available unless a model has been estimated.
- 6, wild bootstrap combined with sieve-AR.

Notes:
1. 0 is not a legitimate option – if set, it is interpreted as 1.
2. If least squares or instrumental variables is the selected estimator, option 1 is automatically changed to 2.
3. The wild bootstrap is not available for Monte Carlo forecasting. If selected, the regular bootstrap will be used instead.

SHOCK_PARS (1×2 row vector of reals): Default = <1,1> .
First element: Variance of Gaussian shocks. Set to 0 to use residual variance
Second element: Skewness factor for wild bootstrap. 1 = symmetric distribution.

RANDNM_SEED (Integer): Default = 0.
Seed for the random number generator. The default setting, 0, causes the actual seed to be generated from the system clock, so that the numbers cannot be replicated.

RANDOM_PRESAMPLE (Boolean): Default = 0.

- 1, Pre-sample Data Random in Simulations
- 0, Pre-sample Data Fixed in Simulations.

TYPEI_FRAC (Boolean): Default = 0.

- 1, Simulate "type I" ARFIMA model
- 0, Simulate regular ARFIMA model.

RESAMPLING (Boolean): Default = 0.

- 1, to compute test *p*-values and standard errors by the parametric bootstrap or subsampling methods.
- 0, for conventional tests.

BOOTSTRAP_REPLICATIONS (Integer): Default = 100.
Number of replications to generate the bootstrap distribution.

BOOTSTRAP_BIASCORR (Boolean) Default = 0.

- 1, to apply bootstrap bias correction to parameter estimates.
- 0, otherwise.

BOOTSTRAP_BLENGTH (Integer) Default = 1.
The length of blocks to resample in the block bootstrap. Setting to 1 yields the regular bootstrap.

BOOTSTRAP_CONFINT (Boolean) Default = 0.

- 0, to report equal tail bootstrap confidence intervals.
- 1, to report percentile-*t* confidence intervals
- 2, to report symmetric percentile-t confidence intervals.

SUBSAMPLING (Boolean) Default = 0.

- 1, to compute confidence intervals and p-values by the subsampling method.
- 0, otherwise.

BOOTSTRAP_BLENGTH (Integer) Default = 1.
The length of blocks to resample in the block bootstrap. Setting to 1 yields the regular bootstrap.

BOOTSTRAP_SIEVELAGS (Integer) Default = –1.

Maximum lag length to use with sieve-AR bootstrap. The default is to set this automatically as function of sample size – see main documentation.

SUBSAMPLE_LENGTH (Integer) Default = 0.
Length, b_T, of the contiguous samples to be used in subsampling analysis. There are then $T - b_T + 1$ samples used to generate the distributions, where T is sample size.

NEWTON_CONV (Double): Default = 0.0001.
Convergence criterion for Newton-Raphson algorithm used for nonlinear bootstrap.

NEWTON_ITERATIONS (Integer): Default = 20.
Maximum number of iterations of Newton-Raphson algorithm used for nonlinear bootstrap.

NEWTON_ALGRTHM (Boolean) Default = 0.

- 1, to use Newton-Raphson algorithm for Monte Carlo replications.
- 0, to use BFGS.

FD_BOOTSTRAP (Boolean): Default = 0.

- 1, to compute the fast-double bootstrap.
- 0. for conventional bootstrap.

The fast double bootstrap is a device aimed at reducing the error in rejection probability due to estimation error. It is not guaranteed to improve performance in all cases, but showing that a test outcome is robust to this setting increases confidence in the result.

8.7 ML and Dynamics Options

STUDDF_ROOT (Integer): Default = 2.
The Student *t* "degrees of freedom" parameter is raised to this power, to represent the actual d.f. of the likelihood. Set > 1 for better numerical stability. By setting a negative value, 0 can represent the Gaussian case (d.f. = ∞).
*Note: start/fixed/test/bound matrices for the Student t parameters have prefix STUDT_*

LOG_SKEWNESS (Boolean): default = 0;

- 1, estimate the logarithm of the Student skewness parameter (ξ)
- 0, otherwise.

MA_FORM (Boolean): Default = 0.

- 1 to report the moving average coefficients as θ_j in $\theta(L) = 1 + \theta_1 L + \ldots + \theta_q L^q$, and GARCH coefficients as β_j in $\beta(L) = 1 + \beta_1 L + \ldots + \beta_r L^r$.
- 0 to report the moving average coefficients as θ_j in $\theta(L) = 1 - \theta_1 L - \ldots - \theta_q L^q$, and GARCH coefficients as β_j in $\beta(L) = 1 - \beta_1 L - \ldots - \beta_r L^r$.

Setting this option to 1 writes the MA coefficients of an ARMA or GARCH with opposite sign to the AR coefficients, relative to the zero-order terms. The default is consistent with the convention in equation (8), and also more natural. For example, in the ARMA(1,1) model equal roots cancel each other out, and in this case the estimates will be equal. Similarly, in the ARMA-in-squares representation of the GARCH(1,1) model, $\beta_1 = \delta_1$ corresponds to $\alpha_1 = 0$, (see GARCH_FORM) and the estimates are again equal in this case.

Note: the starting values must match the convention selected.

RESTRICT_LAGS (Boolean): Default = 1.

- 0, to use the available pre-sample data to form lags, when START_SAMPLE > 1.
- 1, to allow truncation of presample lags.

LAG_TRUNCATION (Integer):Default = 0.

- The maximum number of pre-sample observations to be used to form lags. If set to zero, the estimates are comparable to the case when START_SAMPLE = 1. This option is ignored unless RESTRICT_LAGS = 1

TPI_DEE (Boolean): Default = 0.

- 1, Estimate "type I" ARFIMA model.
- 0, Estimate regular ARFIMA model.

GARCH_FORM (Boolean): Default = 0.

- 1 to report the GARCH model in the conventional (Bollerslev, 1986) style, so that the coefficients of lagged squared errors are the coefficients of $\beta(L) - \delta(L)$ in equations (41) and (42).
- 0 to report the coefficients of the "ARMA-in-squares" representation of the GARCH model, in which $\delta(L)$ and $\beta(L)$ are the AR and MA components respectively.

GARCH_INTFORM (Boolean): Default = 0.

- 1 to report the GARCH intercept in Type 2 form, κ in equations (41) and (42).
- 0 to report the GARCH intercept in Type 1 form, ω in equations (41) and (42).

GARCH_INT_POWER (Integer): Default = 1.

The GARCH intercept can be close to 0, the boundary of the parameter space, which can give difficulty to the search algorithm. Estimating the square root or fourth root of the parameter (set to 2 or 4) may resolve a convergence problem. However, be careful to set the starting value, and interpret the estimate, appropriately.

GM_ITS (Integer): Default = 20.

The GARCH-M likelihood is computed by Gauss-Seidel iteration of equations (1) and (41), (or (1) and (42)). This option sets the maximum number of iterations.

GM_H_BOUND (Real): Default = 10.

To stabilise the GARCH-M calculations, h_t (or $h_t^{1/2}$) is trimmed before inclusion in (1). The upper bound is set to GM_H_BOUND times the sample variance (or standard deviation) of the data. Try reducing this setting in case of failure of the algorithm.

ITERATE_EGARCH (Boolean): Default = 1.

- 0, to evaluate the likelihood by the direct nonlinear recursion of equation (42).
- 1, to evaluate the likelihood by Gauss-Seidel interation; i.e., repeatedly solving the linear difference equation obtained by conditioning on $h_t^{-1/2}$ in equation (42).

There may be small differences between the two estimates due to treatment of initial conditions. The main consideration in this choice is one of speed. The nonlinear equation cannot be solved using Ox's vector manipulation capability, and has to be programmed as a loop. This could result in extremely long solution times in large samples, and the iterative method may be quicker.

EGARCH_ITS (Integer): Default = 20.

The maximum number of iterations in the Gauss-Seidel solution of equation (42).

8.8 Optimization and Run Options

MINIMAND (Boolean): Default = 1.

- 1 to report criterion function as minimand
- 0 to report criterion function as maximand

GRID_POINTS (Integer): Default = 0.

Number of grid points to plot in each direction

Note: # function evaluations = GRID_POINTS in 1-dimensional plot and (GRID_POINTS)^2 in 2-dimensional plot.

MAX_ITERATIONS (Integer): Default = 1000.

Maximum number of BFGS iterations.

PRINT_ITERATIONS (Integer): Default = 0.

Frequency to print current position in search. Set to 0 for no printing.

STRONG_CRITERION (Real): Default = –1.

Criterion for strong convergence (see Ox documentation). Default keeps the Ox default.

WEAK_CRITERION (Real): Default = –1.

Criterion for weak convergence (see Ox documentation). Default keeps the Ox default.

ANAL_DERIV (Boolean): Default = 1.

- 1, Use analytic derivatives where available for BFGS iterations.
- 0, Force use ofs numerical derivatives.

SIM_ANNEALING (Boolean): Default = 0.

- 1, enable simulated annealing as a preliminary search algorithm, to provide initial values for BFGS
- 0, to disable simulated annealing

SA_SETTINGS (1 × 4 vector of Integer/Real): Default = <500, 5, 0.85, 10>

These are the adjustable settings for the SA algorithm. They are:

1. Maximum number of SA iterations before switching to BFGS
2. Initial temperature
3. Temperature reduction factor
4. Number of iterations befpre temperature reduction.

For more details on choosing these settings see

> Goffe, William L., Gary D. Ferrier, and John Rogers (1994). Global Optimization of Statistical Functions with Simulated Annealing. *Journal of Econometrics*, 60(1/2): 65-99.

SCALE_FACTOR (Real): Default = 0.

Parameter rescaling factor, to set similar orders of magnitude based on starting values. The default switches this feature off.

CONSTRAIN_OPT (Real): Default = 3.

A penalty is added to the criterion function when dynamic parameters (ARMA and GARCH coefficients and *d*s) exceed the absolute value specified. Constraining the parameter space makes the search routine more robust, and avoids problems with (e.g.) inverted moving average roots. To remove the constraint, set to a large value. Note that intercepts and regression coefficients are not constrained.

REPORT_FAILURE (Boolean): Default = 0.

- 1 to report results for the best point found when the BFGS algorithm fails to converge.
- 0 to report no results, and replace current parameter values with defaults.

Accessing Results

If ACCESS_RESULTS = 1, estimation results are written to program variables instead of to console output. When this option is selected, the following global variables contain the results of the last call to Run_Estimation().

CONVERGENCE_STATUS	(Integer)
CRITERION	(Real)
SELECTION_CRITERIA	(Vector of Real)
PARAMETER_NAMES	(Array of strings)
PARAMETERS	(Vector of Real)
STANDARD_ERRORS	(Vector of Real)
COVARIANCES	(Matrix of Real)
RESIDUAL_VARIANCE	(Real)
DIAGNOSTICS	(Array of Real Matrices)
TESTS	(Matrix of real)
WALD_ STATISTIC	(Real)
LM_STATISTIC	(Real)
RESIDUAL_CORRELOGRAMS	(Matrix of Real)
FORECASTS	(Matrix of Real)
DATA_NAMES	(Array of strings)
DATA_SET	(Matrix of Real)

CONVERGENCE_STATUS is the value returned by the Ox optimization routine, (MaxBFGS or MaxNewton). See the Ox manual for interpretation.

CRITERION is the final value of the estimation criterion. Note that the sign (maximand or minimand) can be optionally changed, see the MINIMAND option in Optimization Options.

To locate elements of the vectors PARAMETERS and STANDARD_ERRORS, and also of the array PARAMETER_NAMES, use the globally defined vector g_cP and defined constants, as follows:

g_cP[UF]	User function parameters
g_cP[IN]	intercept
g_cP[RG1]	regressors of Type 1
g_cP[RG2]	regressors of Type 2
g_cP[RG3]	regressors of Type 3
g_cP[D]	ARFIMA *d*
g_cP[AR]	AR coefficients
g_cP[MA]	MA coefficients
g_cP[BAR]	Bilinear AR coefficients
g_cP[BMA]	Bilinear MA coefficients
g_cP[GI]	GARCH intercept, or Error Variance
g_cP[GAR]	GARCH AR coefficients
g_cP[GMA]	GARCH MA coefficients

g_cP[FGD]	FIGARCH *d* or HYGARCH memory and amplitude parameters
g_cP[TG]	ASYMM coefficient
g_cP[GR1]	GARCH regressors of Type 1
g_cP[GR2]	GARCH regressors of Type 2
g_cP[GR3]	GARCH regressors of Type 3
g_cP[NT]	Student's *t*-degrees of freedom parameter
g_cP[MKS]	Markov switching parameters
g_cP[ES1]	Explaining regime 1
g_cP[ES2]	Explaining regime 2
g_cP[ES3]	Explaining regime 3
g_cP[EQL]	Equilibrium relations
g_cP[COV]	Error correlations (equation systems only)

Example 1: if an AR(3) is fitted, the AR coefficients are found at locations PARAMETERS[g_cP[AR]], PARAMETERS[g_cP[AR]+1] and PARAMETERS[g_cP[AR]+2].

Example 2: If there are two regressors of Type 1 included, the coefficients are at locations
PARAMETERS[g_cP[RG1]] and PARAMETERS[g_cP[RG1]+1]

The names of these parameters can be found in the corresponding elements of the array PARAMETER_NAMES.

If a Markov-switching model is fitted, use the function LocTP.

```
LocTP(const iReg, const iPar)
```

iReg:	the regime
iPar:	the parameter pointer, as defined above
return value:	location in the vector.

If no Markov switching is specified, this function returns its second argument.

Example 3: If an AR(2) with two regimes is fitted, the parameters for each regime are respectively at locations
PARAMETERS[LocTP(0, g_cP[AR])], PARAMETERS[LocTP(0, g_cP[AR]+1)], and PARAMETERS[LocTP(1, g_cP[AR])], PARAMETERS[LocTP(1, g_cP[AR]+1)]

Example 4: In a 3-regime model, the Markov transition probability parameter t_{12} (see (27) of the main document) is located at
PARAMETERS[LocTP(0, g_cP[MKS]+1)]

In multi-equation models, the parameters for each equation are located using the function LocP .

```
LocP(const iEq, const iPar)
```

iEq:	the equation

`iPar`:	parameter block pointer
return value:	location of parameter block in the vector.

Remember that all equations in the system nominally have the same structure, although some parameters may be suppressed by fixing them at 0. VAR and MA parameters are arranged in the order, variables, then lags. In a VAR(2), the coefficients in equation *j* are ordered as AR1(*j*,1), AR1(*j*,2), AR2(*j*,1), AR2(*j*,2), for *j* = 1,2

Example 5: To locate the parameter AR2(2,2) the reference is `PARAMETERS[LocP(1, AR)+3]` (counting equations from zero, note!).

The `LocP` and `LocTP` functions are straightforwardly combined in a Markov-switching system.

Example 6: To locate the parameter in Example 5 for Regime 2 in a Markov-switching VAR, the reference is `PARAMETERS[LocTP(1, LocP(1, AR)+ 3))]`.

Note that parameter groups from `D` to UF inclusive are defined for each equation, whereas parameter groups from `NT` to `COV` inclusive are defined for the system as a whole. These are accessed just as for a single equation model.
For elements of the covariance matrix, use the same system to locate the required row and column of `COVARIANCES`.

`DIAGNOSTICS` is an array, whose elements correspond to equations of the model. Each element consists of a 2×9 matrix, whose columns correspond to the following list. The first rows contain the statistics, while the second rows contain the numbers of degrees of freedom (restrictions under test) associated with each test, in the case of test statistics. For the first six elements (descriptive statistics) the second row elements are zeros. The right-hand column of Table 1 shows the identifiers for the columns. For example, `DIAGNOSTICS[0][SSQ]` constains the residual sum of squares.

0.	Residual Sum of Squares	`SSQ`
1.	Coefficient of Determination (R^2)	`RSQ`
2.	Residual Standard Deviation	`JB1`
3.	Residual Skewness	`JB2`
4.	Residual Kurtosis	`JB3`
5.	Residual Jarque-Bera statistic	`JB4`
6.	Residual *Q* statistic	`BP1`
7.	Squared-Residual *Q* statistic	`BP2`
8.	First ex-post forecast test	`FC1`
9.	Second ex-post forecast test	`FC2`
10.	Durbin Watson statistic	`DWT`
11.	KPSS Statistic	`KPS`
12.	Lo's R/S Statistic	`LRS`
13.	HML Statistic	`HML`

Table 1. Locations of Equation Diagnostics

Note: Items 1-6 are computed from the variance-adjusted residuals in models where the conditional variance is non-constant (GARCH or regime switching).

`TESTS` is a 2×24 matrix whose first row contains test statistics, or zeros if the test is not specified. The second row contains a test parameter which in most cases, where the statistics are asymptotically chi-squared, is the degrees of freedom of the test (the

numerator degrees of freedom if the *F* versions are specified). If there is no optional parameter the second element contains 0. The right-hand column of the table shows the identifiers for the columns. In the case of the ADF test, `TESTS[0][ADF]` contains the ADF test statistic, if this option has been specified, but `TESTS[1][ADF]` contains the number of lags used to compute the statistic, not the degrees of freedom of the test.

0.	Sargan test (IV estimates only)	`SRT`
1.	Durbin-Wu-Hausman test (IV estimates only)	`DWH`
2.	Phillips-Perron cointegration test	`PPC`
3.	Phillips-Perron cointegration test with trend	`PPT`
4.	Augmented Dickey-Fuller cointegration test	`ADF`
5.	Augmented Dickey-Fuller cointegration test with trend	`ADT`
6.	Score test for autocorrelation	`SC1`
7.	Score test for neglected ARCH	`SC2`
8.	Score test for nonlinear functional form	`SC3`
9.	Score test for heteroscedasticity	`SC4`
10.	Score test for autoregressive common factors	`SC5`
11.	Conditional moment test for autocorrelation	`MT1`
12.	Conditional moment test for neglected ARCH	`MT2`
13.	Conditional moment test for nonlinear functional form	`MT3`
14.	Conditional moment test for heteroscedasticity	`MT4`
15.	Conditional moment test for autoregressive common factors	`MT5`
16.	Information matrix test	`IMT`
17.	Nyblom-Hansen specification test	`HLC`
18.	Wald test of specified restrictions	`WDT`
19.	LM test of “Fixed parameter” restrictions (not linear regression)	`LMT`
20.	LM test of specified added regressors	`SLT`
21.	M/CM test of specified added regressors	`SMT`
22.	Kolmogorov-Smirnov test (wild bootstrap distribution	`WBK`
23,…	User-programmed tests (j = 0,1,2…)	`SUT + j`

Table 2. Locations of Test Statistics

For a single equation model, `RESIDUAL_CORRELOGRAMS` is a matrix with four columns, and number of rows equal to `CORRELOGRAM_ORDER`. The columns are: [0] the residual autocorrelation coefficients for each lag, [1] Box-Pierce or Ljung-Box statistics, [2] and [3], same for the squared residuals. For a model with *N* equations, `RESIDUAL_CORRELOGRAMS` has 4*N* columns, where the first *N* columns contain the correlograms for each equation, columns $N + 1$ to $2N$ contain the B-P or L-B statistics, and similarly for the squared residuals. For example, the correlogram for the second equation would be the column vector `RESIDUAL_CORRELOGRAMS[][4]`, where columns are counted from zero, note.

`FORECASTS` is an array with two elements, where the first element contains the level forecasts, and the second the variance forecasts. Each of these elements is itself an array of *N* elements in a model with *N* equations, containing the relevant components for each equation. The form of the matrices contained in these array elements depends on the type of forecast specified. For analytic forecasts, the level forecasts are contained in a matrix with `FORECAST_STEPS` rows and two columns, containing the point forecasts and standard errors respectively. Except in GARCH and Markov-switching variance models, the second element of `FORECASTS` is an empty array. In those cases

its elements contain a single column, the conditional variance forecasts (no standard errors available for these). For Monte Carlo forecasts the level forecasts consist of three columns, containing respectively the medians, the 2.5% percentiles and the 97.5% percentiles. The second set of elements are either empty or, for GARCH and Markov-switching variance models, have three columns, similarly.

For example: in a single-equation model, the point analytic forecasts, or median forecasts, are located in the column vector ((FORECASTS[0])[0])[][0]. The variance forecasts (if any) are located in ((FORECASTS[1])[0])[][0]. Note that a nested array is defined even in the single equation case. Be careful also not to confuse the different ways that correlograms and forecasts are arranged by equation! This just reflects the way the calculations are organized in each case.

DATA_SET and DATA_NAMES contain the data matrix and column headings, augmented by any new series retrieved from the run, such as residuals, conditional variances, and simulations.
To access retrieved series, use the LocVar() function in conjunction with the following name patterns. In each case, [j] represents a counter, by default 1,2,3,…, which is incremented at each call of Run_Estimation(). This number can be initialized by setting RUN_ID at the start of your program. [i] must be replaced by a number between 1 and $M - 1$, denoting the regime. Successive simulations retrieved following Run [j] are labelled by [k] = 1,2,3,… (Do not include the [] in any of these identifiers.)

Residuals[j]	.	Residuals.
VarAdjResids[j]	.	Residuals divided by Conditional SDs.
ConditionalVars[j]	.	Conditional Variances.
Rg[i]_FilProbs[j]	.	Filter probabilities for Regime *i*.
Rg[i]_SwProbs[j]	.	Explained switch probabilities for Regime *i*.
SmTrWeights[j]	.	Smooth Transition Weights (G_t).
Simulation[j]_[k]	.	Simulation.

For example, the Ox statement

```
decl res = DATA_SET[][LocVar("Residuals1")];
```

places the retrieved residuals from the first run into the variable res.

In multiple equation models, the identifiers receive a suffix of the general form "Eq[m]_[j]" for the series from the *m*th equation in the system. For example, the residuals from Equation 1 on Run 1 are retrieved by a statement such as

```
decl res1 = DATA_SET[][LocVar("ResidualsEq1_1")];
```

TSM Graphics Reference

Graphics Functions

The following TSM functions creates a graphic or graphics, These are either displayed on the monitor using Gnuplot, or saved as a file in one of a range of bitmat and vector graphic formats.

```
Make_Graphic (const iType, const vPlotCode, const
      vVarlist, const vFlags, const bExport)
```

Creates a graphic, or graphics. No return value.

`iType` (integer): The category of plot.

0: Equation-related series.
1: Data series.
2: Recursive estimation parameters and statistics.
3: Bootstrap frequency distributions.
4: Monte Carlo frequency distributions.
5: Selected distribution plots from MC experiments.
6: Tabulated density plots.
7: EDF plots.

`vPlotCode` (vector of integers): The type(s) of plot to be produced for data or equation outputs. If `iType` > 1 this argument is ignored; set it to 0.

Case `iType` = 0: Elements drawn from the values shown in Table 3. .
Case `iType` = 1: Elements can be drawn from the following values.

0: Series plot.
1: Correlogram
1: Partial correlogram.
2: Spectrum
3: Normal QQ plot
4: Histogram/Kernel density
5: Scatter plot (2 series)
6: Bivariate histogram/kernel density (2 series, 3D plot)

Case `iType` > 1: Not used, set to 0.

`vVarlist` The format of this argument depends on the category of plot.

Case `iType` = 0: set to 0 (not used).
Case `iType` = 1: *either* a row *k*-vector of integers *or* a $3 \times k$ matrix of integers.
The (top) row contains numbers of columns of the data matrix. The second and third rows, where present, may contain the line style information for plotting: repectively, the colour/ monochrome pattern index, and the width index. If these rows are not present, the default sequence of line colours is used, with the default line width.
Case `iType` > 1: a row *k*-vector of integers.
These contains the locations of distributions relating to

parameter estimates and statistics. Export the outputs as a spreadsheet to determine their location.

`vFlags`

The definition of of this argument depends on the type of plot.

Case `iType` = 0: a 1 × 2 row vector of integers.

This argument controls the plotting of output for multiple equation models.

`vFlags[0]`:

i, to plot *i*th equation output only.

–1, all if equation outputs to be plotted, as multiple graphs in the same frame.

–2, if all equation outputs to be plotted in a single graph.

`vFlags[1]`:

0 unless the model is an ECM or VECM.

i, to plot *i*th equilibrium relation.

–1, if equilibrium relations to be plotted as multiple graphs in the same frame.

–2, if equilibrium relations to be plotted in a single graph.

Case `iType` = 1: a 1 × 6 row vector of Boolean.

`vFlags[0]`: 1 = detrend the data series by regression
`vFlags[1]`: 1 = difference the data series
`vFlags[2]`: 1 = centre the data series
`vFlags[3]`: 1 = standardize the data series
`vFlags[4]`: 1 = multiple series plot
`vFlags[5]`: 1 = use right-hand scale for last series.

Case `iType` = 5: Boolean, Default = 0.

- 1, Display the selected kernel density plots (if more than one) in a single graph.
- 0, Display the selected kernel density plots as separate graphs in in a single frame.

Cases `iType` = 2,3,4,6,7: Not used, set to 0.

`bExport` (Integer) Default = 0.

- 0, Display plot on the monitor, with Gnuplot.
- 1, Create graphics file with specified format

`Nonparametric_Regression (const Series, const Regressor, const iBand, const bScatter)`

Plots the Nadaraya –Watson bivariate regression curve. No return value.

`Series` (String): Name of the dependent variable, *or* (Integer) Column number of the series.
`Regressor` (String): Name of the regressor variable, *or* (Integer) Column number of the regressor.

`iBand` (Integer): Default = 8.

Bandwidth for Gaussian smoothing kernel – minimum value 1, maximum value 26. (Choose desired setting by inspection).

`bScatter` (Boolean): Default = 0.
1, Show scatter plot,
0, Otherwise.

0	Actual and fitted values, time plot.
1	Actual and fitted values, scatter plot.
2	Residuals and ex post forecasts, time plot.
3	Variance-adjusted/generalized residuals, time plot.
4	Conditional variances, time plot.
5	Residual correlogram.
6.	Variance-adjusted/generalized residuals, correlogram.
7.	Residuals, spectrum.
8.	Variance-adjusted/generalized residuals, spectrum.
9.	Absolute residuals, correlogram.
10.	Absolute variance-adjusted/generalized residuals, correlogram.
11.	Absolute residuals, spectrum.
12.	Absolute variance-adjusted/generalized residuals, spectrum.
13.	Residuals, histogram and/or kernel density.
14.	Absolute variance-adjusted/generalized residuals, histogram.
15	Residuals, normal QQ plot.
16	Absolute variance-adjusted/generalized residuals, normal QQ plot.
20	Forecasts and confidence bands, time plot.
21.	Conditional variance forecasts, time plot.
22.	Monte Carlo forecast, frequency plot for selected period.
23.	Monte Carlo conditional variance forecast, frequency plot for selected period.
24	Impulse responses (MA coefficients).
25	Conditional variance impulse responses (GARCH models).
30	Equilibrium relation, time plot.
31.	Equilibrium relation, correlogram.
32	Equilibrium relation, spectrum.
33	Equilibrium relation, histogram.
34	Equilibrium relation, normal QQ plot.
40	Markov-switching filter probabilities, time plot.
41	Markov-switching smoothed probabilities, time plot.
42	Explained switching probabilities, time plot.
43.	Smooth transition regime weights, time plot.
50.	Composite equation plot: actual/fitted time plot and scatter, residual time plot and residual histogram.
51.	Criterion plot (2D or 3D).
52.	Stochastic simulation, time plot.

Table 3. Codes for Equation Plots

Graphics Options

`EXTD_ACTFIT` (Boolean): default = 0.

- 1, Include actual/fitted scatter and residual histogram in composite equation plot.
- 0, Show only actual/fitted and residual time plots in composite equation plot.

DENSITY_BANDWIDTH (Integer > 0): default = 13;
Controls the smoothing of kernel density estimates. 1 gives the minimum smoothing, 50 the maximum. Experiment with this setting to get the plot desired.

FILL_PVPLOTS (Boolean): default = 0.
- 1, Use the "fill" style to display probability and conditional variance plots
- 0, otherwise.

START_PLTSAMPLE (Integer): default = 0, read as 1.
First observation for series plots. 0 is read as 1.

END_PLTSAMPLE (Integer) default = 0, read as the last observation available.
Last observation to be used for series plots.

CONFBAND_STYLE (Integer) default = 0.
Confidence interval style for forecasts and recursions.
- 0, no confidence interval shown.
- 1, confidence bands.
- 2, confidence bars
- 3, fan chart.

DATES_IN_PLOTS (Integer) default = 0.
Style for time axis labelling in series plots
- 0, Date labels, if supplied in the data file, otherwise the natural numbers..
- 1, YY.
- 2, YYYY
- 3, MM/YY
- 4. MM/YYYY
- 5. DD/MM/YY
- 6. YY-MM-DD

GRAPH_CDF (Boolean): default = 0;
- 1, Graph the CDF when displaying histogram/kernel density.
- 0, otherwise.

GRAPH_DENS (Boolean): default = 0;
- 1, Graph the kernel density when displaying histogram/kernel density.
- 0, otherwise.

GRAPH_HIST (Boolean): default = 1;
- 1, Graph the histogram when displaying histogram/kernel density.
- 0, otherwise.

GRAPH_FONT (Integer) default = 0.
Font for titles, legends and axis labelling of graphics (exported grphics files only).
- 0, Arial.
- 1, Times.
- 2, Courier

- 3, Helvetia.

GRAPH_FONTSIZE (Integer) default = 12.

Font size in points for titles, legends and axis labelling of graphics.

GRAPH_NORM (Boolean): default = 1;

- 1, Graph the normal curve with matching moments when displaying histogram/kernel density.
- 0, otherwise.

LEGEND_BOX (Boolean): default = 0;

- 1, Enclose the legend in a box.
- 0, otherwise.

LEGEND_POS (Integer) default = 0.

Legend position in the graph.

- 0, Top left.
- 1, Top right.
- 2, Bottom left
- 3, Bottom right.
- 4. No legend

LINE_STYLES (11 × 5 matrix of integer).

The columns define the plotting styles for eight lines (used in sequence to plot series on the same graph), two scatter plots (sample and forecast period data points respectively), and the fan chart for forecasts.

Row 1 indexes colours
Row 2 indexes line styles for monochrome plots
Row 3 indexes symbol types
Row 4 indexes symbol sizes (1 = smallest)
Row 5 indexes line widths (1 = narrowest)

The codes for rows 1-3 are listed in Table 4

MCFORECAST_DENSITY (Integer) Default = 1.

In a Monte Carlo forecast, the forecast period for which the kernel density /histogram of the distribution of simulations is to be plotted. (See options 22 and 23 in Table 3.)

MONO_GRAPHS (Boolean): Default = 0.

- 1, Draw monochrome graphs using the line styles in Table 4.
- 0, Draw colour graphs using the line colours in Table 4.

NO_ZEROAXES (Boolean): Default = 0.

- 1, Omit zero axes in series plots.
- 0, Include zero axes (grey broken lines) in series plots.

	Code	Default column
Colours:		
Red	2	1
Brown	7	2
Mauve,	5	3, 10
Light green	3	4
Blue	4	5, 9
Olive	12	6
Dark Blue	6	7
Blue-green	8	8
Purple	13	-
Yellow	16	-
Dark green	11	-
Black	1	11
Dark gray	10	-
Light gray	14	-
Band Fill	17	-
Monochrome Styles		
Solid	1	1, 6
Dot	5	2, 7
Dash	3	3, 8
Dot-dash	4	4
Dot-dot-dash	6	5
Symbol Styles		
No symbol	5	-
Triangle	8	1, 10
Square	1	2, 9
Diamond	9	3
Circle	4	4
Cross	11	5
Plus	2	6
Star	3	7
Filled triangle	8	8
Filled square	0	-
Filled diamond	10	-
Filled circle	6	-
Fill	12	-

Table 4: Line and Symbol Style Codes.

OUTPUT_GRAPHICS (Integer): Default = 0

Format of exported graphics files – see the user's manual Section 8.1 for details.

- 0, PNG (bitmap) file.
- 1, GIF (bitmap) file.
- 2, EPS (vector graphics) file.

- 3, FIG (vector graphics) file.
- 4. TEX (vector graphics) file.

PLOT_FEATURES (Integer): Default = 0

Style for plotting time series

- 0, Lines connecting points.
- 1, Symbols marking points.
- 2, Lines and symbols.

PNGCOLS (Integer): Default = 640.

Number of horizontal pixels (columns) in bitmap files.

PNGROWS (Integer): Default = 480.

Number of vertical pixels (rows) in bitmap files.

PREFORCST_RUN (Integer): Default = 50.

Number of observed pre-forecast data points to be included in a forecast plot.

SCATTER_REGS (Boolean): Default = 1.

- 1, Include the *Y*-on-*X* and *X*-on-*Y* regression lines (displayed as grey broken lines) in a scatter plot
- 0, Otherwise.

SCATTER_RGB (Boolean): Default = 1.

- 1, Use the RGB-syle for a scatter plot, so that the plotted points are colour coded for their position in the sample; red = earliest, blue = latest.
- 0, Otherwise.

SET_GRACORRDER (Boolean): Default = 0.

- 1, Use the setting of SUMMSTAT_CORRELS to control the order of plotted correlograms and partial correlograms.
- 0, Use the default (sample size/2) for correlogram orders.

GUI Commands

The following commands control aspects of the GUI, and graphics functions. They can be set in a text input function in the GUI run file, used as described in Appendix E

```
ASSIGNED_BUTTON1     MODEL_AUTCOMMENT     SAVE_DIALPOS
ASSIGNED_BUTTON2     MODEL_PRCOMMENT      SAVE_SETTINGS
BOOTSTRAP_ONCE       MODEL_GENERIC        SAVING_PROMPTS
BUFFER_SIZE          NPR_BANDWIDTH        SELECTION_FILE
DEGFREE_1            NPR_SCATTER          SHARE_OPTPOS
DEGFREE_2            OMIT_DATADESCR       SHOW_FORECASTS
DELETE_TEMPFILES     OUTPUT_DATA          SHOW_TOOLTIPS
DRAG_SELECT          OUTPUT_FILE          STORED_MCPLOTS
ENABLE_NONLIN        RESTORE_DIALOGS      TEXT_SIZE
ERROR_RECOVERY       RETRIEVE_ONCE
LKUP_DISTRIB         SAVE_CURRLISTINGS
```

Index of Functions and Variable Names

(TSM Programming Options in Boldface)

0	Actual and fitted values, time plot.
1	Actual and fitted values, scatter plot.
2	Residuals and ex post forecasts, time plot.
3	Variance-adjusted/generalized residuals, time plot.
4	Conditional variances, time plot.
5	Residual correlogram.
6.	Variance-adjusted/generalized residuals, correlogram.
7.	Residuals, spectrum.
8.	Variance-adjusted/generalized residuals, spectrum.
9.	Absolute residuals, correlogram.
10.	Absolute variance-adjusted/generalized residuals, correlogram.
11.	Absolute residuals, spectrum.
12.	Absolute variance-adjusted/generalized residuals, spectrum.
13.	Residuals, histogram and/or kernel density.
14.	Absolute variance-adjusted/generalized residuals, histogram.
15	Residuals, normal QQ plot.
16	Absolute variance-adjusted/generalized residuals, normal QQ plot.
20	Forecasts and confidence bands, time plot.
21.	Conditional variance forecasts, time plot.
22.	Monte Carlo forecast, frequency plot for selected period.
23.	Monte Carlo conditional variance forecast, frequiency plot for selected period.
24	Impulse responses (MA coefficients).
25	Conditional variance impulse responses (GARCH models).
30	Equilibrium relation, time plot.
31.	Equilibrium relation, correlogram.
32	Equilibrium relation, spectrum.
33	Equilibrium relation, histogram.
34	Equilibrium relation, normal QQ plot.
40	Markov-switching filter probabilities, time plot.
41	Markov-switching smoothed probabilities, time plot.
42	Explained switching probabilities, time plot.
43.	Smooth transition regime weights, time plot.
50.	Composite equation plot: actual/fitted time plot and scatter, residual time plot and residual histogram.
51.	Criterion plot (2D or 3D).
52.	Stochastic simulation, time plot.

Table 3. Codes for Equation Plots