

Nonparametric Time Series Analysis in JMulTi

March 29, 2005

Rolf Tschernig

Univariate nonparametric time series models are a valuable tool for modelling the conditional mean and conditional volatility function of a stochastic process. In JMulTi one can specify, estimate and analyze nonparametric time series models and use them for forecasting.

The following chapter briefly describes all nonparametric time series models that are available in JMulTi and how to use JMulTi to fit their conditional mean and conditional volatility functions. The set of models also includes several seasonal models that differ in their degree of seasonal flexibility.

This chapter consists of two main parts. The first part describes the equations underlying the computations for the various models and modelling stages where each of the Sections 3 to 6 are devoted to a specific nonparametric model. In the second part the user is guided through the nonparametric modelling process using the various JMulTi panels in the nonparametric module. Each of the Sections 8 to 12 is devoted to one panel.

For a discussion of underlying assumptions and the derivation of relevant properties one may e.g. see the introduction to nonparametric time series modelling in Chapter 7 of Tschernig (2004).

1 Available Models

The nonparametric modelling module in JMulTi covers exclusively models for univariate stochastic processes $\{y_t\}_{t \geq 1}$. In addition to the basic nonparametric heteroskedastic nonlinear autoregressive (NAR) model there are three seasonal extensions: the seasonal nonlinear autoregressive (SNAR) model, the seasonal dummy nonlinear autoregressive (SDNAR) model, and the seasonal shift nonlinear autoregressive (SHNAR) model. A definition of each model is given in Section 1.1.

During the modelling process and for model evaluation, it can be useful to additionally employ linear lag selection methods. This can be done in JMulTi for the linear counterparts of the above mentioned models. These include: the autoregressive (AR) model, the periodic autoregressive (PAR) model, the seasonal dummy linear autoregressive (SDAR) model and the seasonal shift linear autoregressive (SHAR) model, see Section 1.2 for their definition.

Before beginning a nonparametric analysis, one should be aware that all nonparametric methods that are implemented in JMulTi are developed for stochastic processes that are stationary and show a quick enough decay of stochastic dependence between observations with an increasing time span. Technically speaking, the stochastic process has to be β -mixing. Thus, one has to remove deterministic or stochastic trends from a time series

prior to nonparametric modelling. Appropriate testing methods can be found in the **JMulTi** module **Initial Analysis** .

1.1 Nonlinear Models

Heteroskedastic nonlinear autoregressive (NAR) model Choosing a NAR model, one assumes that the stochastic process $\{y_t\}$ is generated by the heteroskedastic nonlinear autoregressive (NAR) model

$$y_t = \mu(x_t) + \sigma(x_t)\xi_t \quad (1)$$

where $x_t = (y_{t-i_1}, y_{t-i_2}, \dots, y_{t-i_m})'$ is the vector of all m correct lagged values, $i_1 < i_2 < \dots < i_m$, and the ξ_t 's, $t = i_m + 1, i_m + 2, \dots$, denote a sequence of i.i.d. random variables with zero mean and unit variance. The functions $\mu(x_t)$ and $\sigma(x_t)$ denote the conditional mean and volatility function, respectively. **JMulTi** also allows the conditional volatility function $\sigma(\cdot)$ to have lags different from those of the conditional mean function, see Section 3.3. Note, however, that asymptotic properties of nonparametric estimation and lag selection methods for the conditional mean function only have been derived for the case that the lag vector of $\sigma(\cdot)$ is a subvector of the lag vector of the conditional mean function.

In order to represent seasonal processes it is convenient to replace the time index t by $t = s + S\tau$ where $s = 1, 2, \dots, S$ denotes the season and $\tau = 0, 1, \dots$ represents a new time index.

Seasonal nonlinear autoregressive (SNAR) model Selecting a SNAR model, one assumes that the stochastic process $\{y_t\}$ is generated by the seasonal nonlinear autoregressive (SNAR) model given by

$$y_{s+\tau S} = \mu_s(x_{s+\tau S}) + \sigma_s(x_{s+\tau S})\xi_{s+\tau S} \quad (2)$$

where x_t is as before the vector of all correct lagged values. In contrast to the standard nonlinear autoregression model (1) the regression functions $\{\mu_s(\cdot)\}_{s=1}^S$ are allowed to vary with the S seasons. This is the most general seasonal process available in **JMulTi** and is also a specialization of the PAR model (6).

It has to be stressed that for both the SDNAR and the PAR model the number of observations available for estimating each function is about $T_S \approx T/S$. If this is too small, then one should select a model with restricted seasonality such as the SDNAR or SHNAR model below. While there is no clear cut minimum for T_S , $T_S = 30$ years of quarterly data are certainly not enough.

Seasonal dummy nonlinear autoregressive (SDNAR) model Choosing a SDNAR model, one assumes that the stochastic process is generated by a seasonal dummy nonlinear autoregressive model (SDNAR model)

$$y_{s+\tau S} = \mu(x_{s+\tau S}) + b_s + \sigma(x_{s+\tau S})\xi_{s+\tau S} \quad (3)$$

where the b_s , $s = 1, \dots, S$, denote seasonal parameters. To guarantee identification one defines $b_1 = 0$. This model is a restricted SNAR model since $\mu_s(\cdot) = \mu(\cdot) + b_s$, $s = 1, 2, \dots, S$. The seasonal variation of the functions between the s -th and the 1-th season is restricted to the constant shifts b_s . The SDNAR model generalizes the SDAR model (7).

Seasonal shift nonlinear autoregressive (SHNAR) model If one selects a SHNAR model, one assumes that the stochastic process $\{y_t\}$ is generated by the seasonal shift nonlinear autoregressive model (SHNAR) model

$$y_{s+\tau S} - \delta_s = \mu(y_{s+\tau S-i_1} - \delta_{\{s-i_1\}}, \dots, y_{s+\tau S-i_m} - \delta_{\{s-i_m\}}) + \sigma(y_{s+\tau S-i_1} - \delta_{\{s-i_1\}}, \dots, y_{s+\tau S-i_m} - \delta_{\{s-i_m\}}) \xi_{s+\tau S} \quad (4)$$

where δ_s , $s = 1, \dots, S$ denote seasonal mean shifts and where $\{a\}$ is defined as

$$\{a\} = \begin{cases} S & \text{if } a \text{ modulo } S = 0, \\ a \text{ modulo } S & \text{otherwise.} \end{cases}$$

For identifiability one assumes $\delta_1 = 0$. This is another way of restricting the seasonal nonlinear autoregression model (2) since it is assumed that the seasonal process is additively separable into a seasonal mean shift δ_s , $s = 1, 2, \dots, S$, and a nonseasonal nonlinear autoregression $\{z_t\}$, i.e. $y_{s+\tau S} = \delta_s + z_{s+\tau S}$.

The SHNAR model is another way of generalizing the SDAR model (7).

1.2 Linear Models

Commonly, linear autoregressive models are stated with all lags included up to order p . Since for all nonlinear models the lag selection procedures allow to consider a subset of lags, we state all linear models in a more general way that allows for nonconsecutive lags.

Linear autoregressive models (AR) models Selecting an AR model, one assumes that the stochastic process $\{y_t\}$ is generated by the linear autoregressive model (AR model)

$$y_t = \sum_{j=1}^m \alpha_{i_j} y_{t-i_j} + \epsilon_t. \quad (5)$$

Periodic autoregressive (AR) models Selecting a PAR model, one assumes that the stochastic process $\{y_t\}$ is generated by the periodic AR (PAR) model

$$y_{s+\tau S} = b_s + \sum_{j=1}^m \alpha_{i_j} y_{s+\tau S-i_j} + \epsilon_{s+\tau S}, \quad (6)$$

see, for example, Lütkepohl (1991, Chapter 12).

Seasonal dummy autoregressive (SDAR) models Selecting a SDAR model, one assumes that the stochastic process $\{y_t\}$ is generated by the seasonal dummy linear autoregressive model (SDAR model)

$$y_{s+\tau S} = b_s + \sum_{j=1}^m \alpha_{i_j} y_{s+\tau S-i_j} + \epsilon_{s+\tau S}. \quad (7)$$

SHAR Models A SHAR model is just another representation of SDAR model (7) and is given by

$$y_{s+\tau S} - \delta_s = \sum_{j=1}^m \alpha_{i_j} (y_{s+\tau S-i_j} - \delta_{\{s-i_j\}}) + \epsilon_{s+\tau S} \quad (8)$$

where the constant seasonal mean shifts $\delta_1, \dots, \delta_S$ are obtained up to an additive constant via the system of linear equations $b_s = \delta_s - \sum_{j=1}^p \alpha_j \delta_{\{s-j\}}$, $s = 1, 2, \dots, S$.

2 The Model Building Steps

2.1 Overview

In general, the (nonparametric) modelling process of a univariate time series consists of the following steps:

1. **Prior Data Transformations**
2. **Model and Lag Selection**

3. Estimation

4. Model Checking

and potentially

5. Volatility Analysis

6. Forecasting

Except for the first step each step corresponds to one panel in the **JMulTi** module **Non-parametric Time Series Analysis**. The use of each panel will be explained in Sections 8 to 12. For the formulas and algorithms underlying the computations of each modelling step see Sections 3, 4, 5, and 6 for the NAR, SNAR, SDNAR, and SHNAR model, respectively.

2.2 Prior Data Transformations

Since data transformations are also necessary before fitting standard ARMA models, there is no specific panel for prior data transformations in this model.

The first step in nonparametric modelling is to transform the time series such that it becomes stationary and β -mixing. The latter implies that the stochastic dependence between observations of different time points decreases fast enough with the distance between observations. This condition is fulfilled for standard linear processes with geometrically decaying autocovariance functions and reasonably well behaved error distributions. For details see e.g. Tschernig and Yang (2000) or Yang and Tschernig (2002) in case of seasonal processes. These conditions guarantee asymptotically valid results.

Practically, this implies e.g. a logarithmic transformation and/or removing deterministic or stochastic (seasonal) trends. To check the presence of the latter one can use the panel **Unit Root Tests** in the **JMulTi** module **Initial Analysis**.

3 NAR Models

In this Section we briefly summarize the underlying theory for the NAR model (1). Sections 8 to 12 describe how to carry out the steps of lag selection, estimation, model checking, volatility analysis, and forecasting in **JMulTi**.

3.1 Estimation and Confidence Intervals

In this subsection it is assumed that the relevant lags have already been selected. If the lags are not yet determined, see Section 3.2.

3.1.1 Local Linear Estimation

Let $K(u)$ denote a kernel function and

$$K_h(x_t - x) = \prod_{i=1}^m \frac{1}{h^m} K\left(\frac{x_{ti} - x_i}{h}\right),$$

a product kernel with scalar bandwidth h and with kernel variance $\sigma_K^2 = \int u^2 K(u) du$ and the kernel constant $\|K\|_2^2 := \int K(u)^2 du$. Define further the matrices

$$e = (1, 0_{1 \times m})', \quad \mathbf{Z}(x) = \begin{pmatrix} 1 & \cdots & 1 \\ x_{i_{m+1}} - x & \cdots & x_T - x \end{pmatrix}',$$

$$W(x, h) = \text{diag} \{K_h(x_t - x)/T\}_{t=i_{m+1}}^T, \quad \mathbf{y} = \begin{pmatrix} y_{i_{m+1}} & \cdots & y_T \end{pmatrix}'.$$

For computing the local linear estimator, JMulTi uses the bandwidth

$$\hat{h}_u = c_u \hat{h}_{opt} \tag{9}$$

and a Gaussian kernel where the default setting is $c_u = 1$ and where \hat{h}_{opt} denotes the plug-in bandwidth (24). In the **Specify Estimation Panel**, see Section 9.1, the user may, however, modify this bandwidth by selecting a different factor c_u in the **Factor to multiply plug-in bandwidth with** box. The local linear function estimator for the conditional mean function $\mu(x)$ of a NAR model (1) is given by

$$\hat{\mu}(x, \hat{h}_u) = e' \left\{ \mathbf{Z}'(x) W(x, \hat{h}_u) \mathbf{Z}(x) \right\}^{-1} \mathbf{Z}'(x) W(x, \hat{h}_u) \mathbf{y}. \tag{10}$$

JMulTi computes $\hat{\mu}(\cdot, \hat{h}_u)$ at

- a user given vector x where each element is by default set to the mean of y_t and can be changed in the **Specify Estimation Panel** in the **Evaluate function at** box.
- on a grid of values such that the function can be plotted, see Section 3.1.2, and the **Specify Estimation Panel**, Section 9.1.

3.1.2 Plotting the function

If $m \leq 2$, JMulTi estimates the conditional mean function on a grid of J points $[x^{(1)}, \dots, x^{(J)}]$. If x is scalar, the grid points are equidistantly chosen on the interval $[\min_t y_t, \max_t y_t]$. For a two-dimensional regression vector x , J^2 grid points are used with J equidistant points in direction i covering the interval $[\min_t x_{ti}, \max_t x_{ti}]$, $i = 1, 2$. In the **Specify Estimation Panel**, see Section 9.1, the user can change the default value $J = 5$ in the **Grid points in each direction** box. Since this grid is rectangular, the function $\mu(\cdot)$ is also computed at points which are outside the data range. All function estimates at points outside the data

range are suppressed from the plot since those estimates may exhibit a quite large estimation variance.

Occasionally, the plot is strongly influenced by a few estimates close to the boundary of the sample. If this is the case, then such data points can be expected to have a small density just like points that are completely outside the data range. One therefore may wish to exclude such points from the plot as well. Therefore, one can remove in **JMulTi** from the plot all function estimates at points for which the estimated density $\hat{f}(x, \hat{h}_B)$ is in the b quantile of lowest density values where \hat{h}_B is given by (20). This is done in the same panel in the box **Quantile of grid points to be suppressed in plot**.

To estimate the density $f(x)$ a kernel estimator is used. In order to exploit all available data for estimating $f(x)$, the kernel estimator

$$\hat{f}(x, \hat{h}_B) = (T - i_m + i_1)^{-1} \sum_{t=i_m+1}^{T+i_1} K_{\hat{h}_B}(x_t - x) \quad (11)$$

with the Gaussian kernel and the rule-of-thumb bandwidth (20) is used. The vectors x_t , $t = T + 1, \dots, T + i_1$ are all available from the observations y_t , $t = 1, \dots, T$. For example, x_{T+i_1} is given by $(y_T, \dots, y_{T+i_1-i_m})'$. This robustification which was suggested by Tjøstheim and Auestad (1994) is switched off if the sum stops at T .

If there are more than two lags, $m > 2$, and one wants to plot the function, one lets x vary in the directions of two lags and conditions on the values of the remaining lags. In the **Specify Estimation Panel** the user can specify in the boxes **1st lag to plot** and **2nd lag to plot** for which lags to plot the function. The conditioning values are by default set to the mean of y_t and can be changed in the box **Condition plot at**. When interpreting such plots one has to bear in mind that changing the conditioning values also changes the plot.

3.1.3 Confidence Intervals

By checking in the **Specify Estimation Panel**, see Section 9, the **Plot CIs (no 3D)** box, one can also compute confidence intervals if one allows the function to vary within one direction only. The confidence intervals are computed at the grid points based on

$$\left[\hat{\mu}(x, \hat{h}_u) - z_{\alpha/2} \sqrt{\frac{\hat{v}(x, \hat{h}_{\sigma, opt}, \hat{h}_u, \hat{h}_B)}{T \hat{h}_u^m}}, \quad \hat{\mu}(x, \hat{h}_u) + z_{\alpha/2} \sqrt{\frac{\hat{v}(x, \hat{h}_{\sigma, opt}, \hat{h}_u, \hat{h}_B)}{T \hat{h}_u^m}} \right] \quad (12)$$

where $\hat{h}_u = c_u \hat{h}_{opt}$, z_β denotes the $(1 - \beta)$ quantile of the normal distribution and the estimator of the asymptotic estimation variance is given by

$$\hat{v}(x, \hat{h}_{\sigma, opt}, \hat{h}_u, \hat{h}_B) = \frac{\hat{\sigma}^2(x, \hat{h}_{\sigma, opt}, \hat{h}_u) \|K\|_2^{2m}}{\hat{f}(x, \hat{h}_B)}. \quad (13)$$

Here, the conditional variance estimator $\hat{\sigma}^2(x, \hat{h}_{\sigma, opt}, \hat{h}_u)$, the plug-in bandwidth $\hat{h}_{\sigma, u}$ for the conditional variance estimator and the density estimator $\hat{f}(x, \hat{h}_B)$ are given below by (31), (38), and (11), respectively. In the **CI level** box one can change the default value of 0.95 of the confidence level $1 - \alpha$.

By checking the **Bonferroni CIs** box, one also can compute Bonferroni confidence intervals. Thus, replacing α in (12) by

$$\alpha_j = \alpha/J \tag{14}$$

for each grid point guarantees that the overall confidence level is at least the desired confidence level $(1 - \alpha)$.

In addition to the confidence intervals (12) based on the conditional volatility function $\sigma^2(x)$, JMulTi also computes confidence intervals based on the assumption of homoskedastic errors by replacing in (13) the conditional variance estimator by the variance estimator (26) below.

3.2 Lag and Bandwidth Selection

3.2.1 The formulas

This section describes the nonparametric lag and bandwidth selection methods used in JMulTi . All the settings described below can be controlled in the **Specify Lag Selection** Panel, see Section 8 for a detailed description.

Nonparametric lag selection in JMulTi is based on estimating the Asymptotic Final Prediction Error (AFPE) or a corrected version thereof (CAFPE). For lag selection it is necessary to a priori specify a set of possible lag vectors by choosing the largest candidate lag M in the **Largest Candidate Lag** box. Denote the full lag vector containing all lags up to M by $x_{t,M} = (y_{t-1}, y_{t-2}, \dots, y_{t-M})'$.

Let $w(\cdot)$ denote a weight function. Then for the data generating process with lags i_1, \dots, i_m and bandwidth h the Asymptotic Final Prediction Error is given by

$$AFPE(h, i_1, \dots, i_m) = A + b(h)B + c(h)C \tag{15}$$

where the integrated variance A , the integrated variance of estimation $b(h)B$ and the integrated squared bias of estimation $c(h)C$ with the (unknown) constants are given by

$$\begin{aligned}
A &= \int \sigma^2(x)w(x_M)f(x_M)dx_M = E [\sigma^2(x_t)w(x_{t,M})], \\
B &= \int \sigma^2(x)w(x_M)f(x_M)/f(x)dx_M = E \left[\sigma^2(x_t) \frac{w(x_{t,M})}{f(x_t)} \right], \\
C &= \int \left(\text{tr} \left\{ \frac{\partial^2 \mu(x)}{\partial x \partial x'} \right\} \right)^2 w(x_M)f(x_M)dx_M \\
&= E \left[\left(\text{tr} \left\{ \frac{\partial^2 \mu(x_t)}{\partial x \partial x'} \right\} \right)^2 w(x_{t,M}) \right]
\end{aligned} \tag{16}$$

and

$$b(h) = \|K\|_2^{2m}(T - i_m)^{-1}h^{-m}, \quad c(h) = \sigma_K^4 h^4/4. \tag{17}$$

Note that the last two terms depend on the bandwidth and kernel constants only while A , B and C are functionals of the unknown NAR model. The idea is to choose the lag combination which leads to the smallest $AFPE(\cdot)$.

For a given lag combination $\{i_1, \dots, i_m\}$ the $AFPE(h, i_1, \dots, i_m)$ can also be used to obtain the asymptotically optimal (global) bandwidth simply by minimizing $AFPE(\cdot)$ with respect to h . It is given by

$$h_{opt} = \left\{ \frac{m \|K\|_2^{2m} B}{(T - i_m) \sigma_K^4 C} \right\}^{1/(m+4)}. \tag{18}$$

In practice, the constants A , B and C are unknown and have to be estimated. To maintain the generality of the nonparametric model they are also nonparametrically estimated.

The constant B is estimated by

$$\widehat{B}(\widehat{h}_B) = \frac{1}{T - i_m} \sum_{t=i_m+1}^T \left\{ y_t - \widehat{\mu}(x_t, \widehat{h}_B) \right\}^2 \frac{w(x_{t,M})}{\widehat{f}(x_t, \widehat{h}_B)} \tag{19}$$

where $\widehat{f}(\cdot, \cdot)$ is the Gaussian kernel estimator (11) of the density $f(x)$ and \widehat{h}_B is Silverman's (1986) rule-of-thumb bandwidth

$$\widehat{h}_B = \widehat{\sigma} \left(\frac{4}{m+2} \right)^{1/(m+4)} T^{-1/(m+4)} \tag{20}$$

with $\widehat{\sigma} = \left(\prod_{j=1}^m \sqrt{Var(x_{t,j})} \right)^{1/m}$ denoting the geometric mean of the standard deviation of the regressors.

The constant C is estimated by

$$\widehat{C}(\widehat{h}_C) = \frac{1}{T - i_m} \sum_{t=i_m+1}^T \left[\sum_{j=1}^m \widehat{\mu}^{(jj)}(x_t, \widehat{h}_C) \right]^2 w(x_{t,M}), \quad (21)$$

where $\mu^{(jj)}(\cdot, \cdot)$ denotes the second-order direct derivative of the function $\mu(\cdot)$. The second-order direct derivatives are estimated with the direct local quadratic estimator

$$\begin{aligned} \{\widehat{c}_0, \widehat{c}_{11}, \dots, \widehat{c}_{1m}, \widehat{c}_{21}, \dots, \widehat{c}_{2m}\} = \arg \min_{\{c_0, c_{11}, \dots, c_{1m}, c_{21}, \dots, c_{2m}\}} \\ \sum_{t=i_m+1}^T \{y_t - c_0 - c_{11}(x_{t1} - x_1) - \dots - c_{1m}(x_{tm} - x_m) \\ - c_{21}(x_{t1} - x_1)^2 - \dots - c_{2m}(x_{tm} - x_m)^2\}^2 K_h(x_t - x). \end{aligned} \quad (22)$$

where the estimates of the direct second derivatives are given by $\widehat{\mu}^{(jj)}(x, \widehat{h}_C) = 2\widehat{c}_{2j}$, $j = 1, \dots, m$. The rule-of-thumb bandwidth \widehat{h}_C is calculated as

$$\widehat{h}_C = 2\widehat{\sigma} \left(\frac{4}{m+4} \right)^{1/(m+6)} T^{-1/(m+6)} \quad (23)$$

and has the correct asymptotic order, see Yang and Tschernig (1999).

Based on the estimated B and C , the PLUG-IN BANDWIDTH is given by

$$\widehat{h}_{opt} = \left\{ \frac{m \|K\|_2^{2m} \widehat{B}(\widehat{h}_B)}{(T - i_m) \widehat{C}(\widehat{h}_C) \sigma_K^4} \right\}^{1/(m+4)}. \quad (24)$$

The constant A is estimated by

$$\widehat{A}(h) = \frac{1}{T - i_m} \sum_{t=i_m+1}^T \{y_t - \widehat{\mu}(x_t, h)\}^2 w(x_{t,M}) \quad (25)$$

In order to eliminate the asymptotic first-order bias from $\widehat{A}(h)$, the asymptotic bias is subtracted. Thus, the integrated variance is estimated by

$$\widehat{A}(\widehat{h}_u) - \left\{ b(\widehat{h}_u) - \frac{2K(0)^m}{(T - i_m) \widehat{h}_u^m} \right\} \widehat{B}(\widehat{h}_B) - c(\widehat{h}_u) \widehat{C}(\widehat{h}_C). \quad (26)$$

where as before $\widehat{h}_u = c_u \widehat{h}_{opt}$. Note that if c_u is chosen too large, the last term becomes large too and a negative estimate may result.

Taking all estimates together, one can estimate the $AFPE(\cdot)$ given by (15). Note that for computing the nonparametric lag selection criteria **JMulTi** always uses the plug-in bandwidth \hat{h}_{opt} . The ASYMPTOTIC FINAL PREDICTION ERROR (AFPE) is calculated as

$$AFPE = \hat{A}(\hat{h}_{opt}) + \frac{2K(0)^m}{(T - i_m)\hat{h}_{opt}^m} \hat{B}(\hat{h}_B). \quad (27)$$

Tschernig and Yang (2000) have shown that conducting lag selection based on $AFPE$ is consistent if the underlying data generating process is not linear, that is if C given by (16) is positive.

Despite its consistency Tschernig and Yang (2000) have shown that the $AFPE$ tends to include lags in addition to the correct ones. In other words, overfitting is quite likely. Therefore they suggest a CORRECTED ASYMPTOTIC FINAL PREDICTION ERROR (CAFPE). The CAFPE is calculated as

$$CAFPE = AFPE \{1 + m(T - i_m)^{-4/(m+4)}\}. \quad (28)$$

It is also consistent but exhibits much better finite sample behavior. It should therefore be used in practice.

For the choice of AFPE or CAFPE, the choice of a strategy to search through the possible lag combinations and other issues, see Section 8.1.2.

3.2.2 Implementation

In all computations the weight function $w(\cdot)$ is taken to be the indicator function on the data.

Computing \hat{B} : For estimating the density the leave-one-out version of (11) is used. For computing $\hat{\mu}(x_t, \hat{h}_B)$ the leave-one-out estimator is used as well.

Since function estimates for which the estimated density is within the 5%-quantile of the lowest density values are likely to be unreliable, they are not used for computing B . The number of observations in the denominator is adjusted accordingly.

In practice it may happen that the inversion in (10) fails due to numerical inaccuracies, e.g. if there are only few observations or the bandwidth is relatively small. In this case, **JMulTi** multiplies the current bandwidth by 1.05 and computes (10) with the larger bandwidth. If the inversion still does not work, this procedure is repeated. If no success is reached after 30 iterations, the program stops.

Computing $\hat{C}(\cdot)$: the last two measures are applied as well where (10) is replaced by (22).

3.3 Conditional Volatility Analysis

3.3.1 Estimation of the Volatility Function

If the conditional mean function were known, one can define the 'demeaned' errors

$$\varepsilon_t = y_t - \mu(x_t), \quad t = i_m + 1, \dots, T.$$

Inserting into (1) and manipulating a little bit leads to $\varepsilon_t^2 = \sigma^2(x_t) + \sigma^2(x_t)(\xi_t^2 - 1)$, $t = i_m + 1, \dots, T$. In practice the lags in $\sigma^2(\cdot)$ may differ from those in $\mu(\cdot)$. This is expressed by generalizing (1) and allowing the conditional variance function to depend on a lag vector $x_{\sigma,t}$ that is potentially different from x_t :

$$\varepsilon_t^2 = \sigma^2(x_{\sigma,t}) + \sigma^2(x_{\sigma,t})(\xi_t^2 - 1), \quad t = i_{\sigma,m} + 1, \dots, T, \quad (29)$$

and where $i_{\sigma,m} = \max(i_m, i_{m_\sigma})$. Here i_{m_σ} denotes the largest lag in x_σ .

The stochastic process (29) is a slight modification of the NAR model (1) since the lags in $x_{\sigma,t}$ are not lags of the new dependent variable ε_t^2 . A local linear estimator of $\sigma^2(x_\sigma)$ is obtained by slightly modifying (10).

Since in general $\mu(\cdot)$ and therefore the errors ε_t 's are unknown, the errors have to be estimated by

$$\widehat{\varepsilon}_t(\widehat{h}_u) = y_t - \widehat{\mu}(x_t, \widehat{h}_u), \quad t = i_m + 1, \dots, T. \quad (30)$$

The vector of squared residuals is then denoted by $\widehat{\varepsilon}^2(\widehat{h}_u) = \left(\widehat{\varepsilon}_{i_m+1}^2(\widehat{h}_u), \dots, \widehat{\varepsilon}_T^2(\widehat{h}_u) \right)'$.

In the following all formulas will be given for the case of estimated errors. However, if $\mu(\cdot)$ is known, the vector of squared residuals $\widehat{\varepsilon}^2(\widehat{h}_u)$ is replaced by ε^2 and if $\mu(\cdot)$ is assumed to be zero, then $\widehat{\varepsilon}^2(\widehat{h}_u)$ is replaced by \mathbf{y}^2 .

Using the vector of squared residuals, the local linear estimator of the conditional variance function $\sigma^2(x_\sigma)$ is given by

$$\widehat{\sigma}^2(x_\sigma, \widehat{h}_{\sigma,u}, \widehat{h}_u) = e' \left\{ \mathbf{Z}'(x_\sigma) W(x_\sigma, \widehat{h}_{\sigma,u}) \mathbf{Z}(x_\sigma) \right\}^{-1} \mathbf{Z}'(x_\sigma) W(x_\sigma, \widehat{h}_{\sigma,u}) \widehat{\varepsilon}^2(\widehat{h}_u) \quad (31)$$

where

$$\widehat{h}_{\sigma,u} = c_{\sigma,u} \widehat{h}_{\sigma,opt}. \quad (32)$$

By default $c_{\sigma,u} = 1$ and the plug-in bandwidth (38) is used. Sometimes the plotted function estimate may look not "smooth enough". Then the user may increase the bandwidth $\widehat{h}_{\sigma,u}$

by setting $c_{\sigma,u}$ to a larger value. Note that if $c_{\sigma,u}$ is chosen very large, one obtains almost a straight line that usually is up- or downward sloping.

If the inversion in (31) fails, then the computation is redone after multiplying the previously used bandwidth by 1.05. This is done at most 30 times.

Note that the estimator $\hat{\sigma}^2(x_\sigma, \hat{h}_{\sigma,u}, \hat{h}_u)$ can lead to negative estimates even if a very large bandwidth $\hat{h}_{\sigma,u}$ is used. If such a situation arises, **JMulTi** replaces the local linear estimator (31) by a local constant (Nadaraya-Watson) estimator

$$\hat{\sigma}_{NW}^2(x_\sigma, \hat{h}_{\sigma,u}, \hat{h}_u) = \frac{\sum_{t=i_{\sigma,m}+1}^T K_{\hat{h}_{\sigma,u}}(x_{\sigma,t} - x_\sigma) \widehat{\varepsilon}_t^2(\hat{h}_u)}{\sum_{t=i_{\sigma,m}+1}^T K_{\hat{h}_{\sigma,u}}(x_{\sigma,t} - x_\sigma)}. \quad (33)$$

3.3.2 Bandwidth and Lag Selection

In order to state bandwidth and lag selection for the conditional volatility function one has to adapt the formulas of Section 3.2 with respect to the current dependent variables, the squared residuals (29). Then for the data generating process with lags $i_{\sigma,1}, \dots, i_{\sigma,m}$ and bandwidth h the Asymptotic Final Prediction Error for the conditional variance is given by

$$AFPE_\sigma(h, i_{\sigma,1}, \dots, i_{\sigma,m}) = A_\sigma + b(h)B_\sigma + c(h)C_\sigma \quad (34)$$

where the integrated variance A_σ , the integrated variance of estimation $b(h)B_\sigma$ and the integrated squared bias of estimation $c(h)C_\sigma$ with the (unknown) constants are given by

$$\begin{aligned} A_\sigma &= (m_4 - 1)E \left[\sigma^4(x_{\sigma,t}) w(x_{t,M_\sigma}) \right], \\ B_\sigma &= (m_4 - 1)E \left[\sigma^4(x_{\sigma,t}) \frac{w(x_{t,M_\sigma})}{f(x_{\sigma,t})} \right], \\ C_\sigma &= E \left[\left(\text{Tr} \left\{ \frac{\partial^2 \sigma^2(x_{\sigma,t})}{\partial x_\sigma \partial x'_\sigma} \right\} \right)^2 w(x_{t,M_\sigma}) \right] \end{aligned}$$

and $b(h)$ and $c(h)$ as before and $M_\sigma \geq i_{\sigma,m}$.

The asymptotically optimal bandwidth is given by

$$h_{\sigma,opt} = \left\{ \frac{m_\sigma \|K\|_2^{2m_\sigma} B_\sigma}{(T - i_{\sigma,m}) \sigma_K^4 C_\sigma} \right\}^{1/(m_\sigma+4)}. \quad (35)$$

The constant B_σ is estimated by

$$\widehat{B}_\sigma(\widehat{h}_B, \widehat{h}_u) = \frac{1}{T - i_{\sigma,m}} \sum_{t=i_{\sigma,m}+1}^T \left\{ \widehat{\varepsilon}_t^2(\widehat{h}_u) - \widehat{\sigma}^2(x_{\sigma,t}, \widehat{h}_B, \widehat{h}_u) \right\}^2 \frac{w(x_{t,M_\sigma})}{\widehat{f}(x_{\sigma,t}, \widehat{h}_B)} \quad (36)$$

where $\widehat{f}(\cdot)$ is the Gaussian kernel estimator (11) of the density $f(x_\sigma)$ and \widehat{h}_B is Silverman's (1986) rule-of-thumb bandwidth (20) where the lag vector x_σ , $i_{1,\sigma}$ and m_σ are used. The estimator $\widehat{\sigma}^2(x_{\sigma,t}, \widehat{h}_B, \widehat{h}_u)$ is given by (31).

The constant C_σ is estimated by

$$\widehat{C}_\sigma(\widehat{h}_C, \widehat{h}_u) = \frac{1}{T - i_{\sigma,m}} \sum_{t=i_{\sigma,m}+1}^T \left[\sum_{j=1}^m \widehat{\sigma}^{2(jj)}(x_t, \widehat{h}_C, \widehat{h}_u) \right]^2 w(x_{t,M_\sigma}), \quad (37)$$

where $(\sigma^2)^{(jj)}(\cdot)$ denotes the second-order direct derivative of the function $\sigma^2(\cdot)$. The second-order direct derivatives are estimated with an appropriately modified version of the direct local quadratic estimator (22). Based on the estimated B_σ and C_σ , the plug-in bandwidth for the conditional volatility estimator is given by

$$\widehat{h}_{\sigma,opt} = \left\{ \frac{m_\sigma \|K\|_2^{2m_\sigma} \widehat{B}_\sigma(\widehat{h}_B, \widehat{h}_u)}{(T - i_{\sigma,m}) \widehat{C}_\sigma(\widehat{h}_C, \widehat{h}_u) \sigma_K^4} \right\}^{1/(m_\sigma+4)}. \quad (38)$$

The constant A_σ is estimated by

$$\widehat{A}_\sigma(h, \widehat{h}_u) = \frac{1}{T - i_{\sigma,m}} \sum_{t=i_{\sigma,m}+1}^T \left\{ \widehat{\varepsilon}_t^2(\widehat{h}_u) - \widehat{\sigma}^2(x_t, h, \widehat{h}_u) \right\}^2 w(x_{t,M_\sigma}). \quad (39)$$

In order to eliminate the asymptotic first-order bias from $\widehat{A}_\sigma(h, \widehat{h}_u)$, the asymptotic bias is subtracted. Thus, the integrated variance is estimated by

$$\widehat{A}_\sigma(\widehat{h}_{\sigma,u}, \widehat{h}_u) = \left\{ b(\widehat{h}_{\sigma,u}) - \frac{2K(0)^{m_\sigma}}{(T - i_{\sigma,m}) \widehat{h}_{\sigma,u}^{m_\sigma}} \right\} \widehat{B}_\sigma(\widehat{h}_B, \widehat{h}_u) - c(\widehat{h}_{\sigma,u}) \widehat{C}_\sigma(\widehat{h}_C, \widehat{h}_u), \quad (40)$$

where as before $\widehat{h}_{\sigma,u} = c_{\sigma,u} \widehat{h}_{\sigma,opt}$.

Taking all estimates together, one can estimate the $AFPE_\sigma(\cdot)$. Note that **JMulTi** here always uses the plug-in bandwidth $\widehat{h}_{\sigma,opt}$. The AFPE for the conditional volatility is calculated as

$$AFPE_\sigma = \widehat{A}_\sigma(\widehat{h}_{\sigma,opt}, \widehat{h}_u) + \frac{2K(0)^{m_\sigma}}{(T - i_{\sigma,m}) \widehat{h}_{\sigma,opt}^{m_\sigma}} \widehat{B}_\sigma(\widehat{h}_B, \widehat{h}_u). \quad (41)$$

So far, consistency of lag selection based on (41) has only been shown for the special case of $f(\cdot) = 0$ on the domain of the weight function $w(\cdot)$. However, similar to the case of the conditional mean function, lag selection based on $AFPE_\sigma$ cannot be consistent if the process is homoskedastic since then $C_\sigma = 0$ and no bias-variance tradeoff exists. The latter is required for the existence of an asymptotically optimal bandwidth. Thus, in case of

homoskedastic processes, this lag selection procedure is likely to select superfluous lags! For a non-zero conditional mean function $f(\cdot)$ there do not yet exist consistency results for this lag selection estimator although consistency may be conjectured.

Assuming that $AFPE_\sigma$ also tends to include lags in addition to the correct ones, JMuTI estimates the CAFPE for the conditional volatility. It is calculated as

$$CAFPE_\sigma = AFPE_\sigma \{1 + m_\sigma(T - i_{\sigma,m})^{-4/(m_\sigma+4)}\}. \quad (42)$$

3.3.3 Implementation

The same measures as for the conditional mean function are implemented. See Section 3.2.2.

3.3.4 Residual Analysis for Volatility Estimates

In order to check whether the fitted conditional volatility model is appropriate, the errors ξ_t , $t = i_m + 1, \dots, T$ are estimated by

$$\hat{\xi}_t(\hat{h}_{\sigma,u}, \hat{h}_u) = \frac{\hat{\varepsilon}_t(\hat{h}_u)}{\sqrt{\hat{\sigma}^2(x_t, \hat{h}_{\sigma,u}, \hat{h}_u)}} = \frac{y_t - \hat{\mu}(x_t, \hat{h}_u)}{\sqrt{\hat{\sigma}^2(x_t, \hat{h}_{\sigma,u}, \hat{h}_u)}}, \quad t = i_m + 1, \dots, T, \quad (43)$$

using the estimators presented in Section 3.3.1. All the residual checking for the volatility estimates are done using the $\hat{\xi}_t$'s. Note if the conditional mean function $\mu(\cdot)$ and the conditional volatility function $\sigma(\cdot)$ of the NAR model (1) are correctly specified, the estimated ξ_t 's should look like an i.i.d. sequence with mean 0 and variance 1. See the **Check Residuals of Volatility Estimation Panel** in Section 11.3.

4 SNAR Models

This section summarizes the formulas and settings for the nonparametric estimation of the seasonal nonlinear autoregressive (SNAR) model (2). The estimation and lag selection is facilitated if there is an equal number of observations for each season. One then has to guarantee that one has at least $M \geq i_m$ starting values for each season and estimation has to start in $\tau = i_{M,S}$, where $i_{M,S}$ is the smallest integer equal to or greater than M/S . The largest value of the nonseasonal index τ is given by $T_S = \lceil \frac{T}{S} \rceil - 1$, where $\lceil a \rceil$ truncates the fractional portion of a . The number of observations per season that is available for estimation if the largest lag is M is denoted by $T_{M,S} = T_S - i_{M,S} + 1$.

4.1 Estimation and Confidence Intervals

In this subsection it is assumed that the relevant lags have already been selected. If the lags are not yet determined, see Section 4.2.

4.1.1 Local Linear Estimation

The local estimators for estimating the seasonal conditional mean functions $\mu_s(\cdot)$, $s = 1, \dots, S$, are obtained by estimating each seasonal function separately using only data of season s . The local linear function estimators are given by

$$\hat{\mu}_s(x, \hat{h}_{S,u}) = e' \left\{ \mathbf{Z}'_s(x) W_s(x, \hat{h}_{S,u}) \mathbf{Z}_s(x) \right\}^{-1} \mathbf{Z}'_s(x) W_s(x, \hat{h}_{S,u}) \mathbf{y}_s, \quad s = 1, \dots, S, \quad (44)$$

where the following notation is used:

$$\mathbf{Z}_s(x) = \begin{pmatrix} 1 & \cdots & 1 \\ x_{s+i_{M,S}S} - x & \cdots & x_{s+T_S S} - x \end{pmatrix}',$$

$$W_s(x, h_s) = \text{diag} \{ K_{h_s}(x_{s+\tau S} - x) / T_{M,S} \}_{\tau=i_{M,S}}^{T_S},$$

$$\mathbf{y}_s = \begin{pmatrix} y_{s+i_{M,S}S} & y_{s+(i_{M,S}+1)S} & \cdots & y_{s+T_S S} \end{pmatrix}'.$$

For computing the local linear estimator, JMulTi uses the bandwidth

$$\hat{h}_{S,u} = c_u \hat{h}_{S,opt} \quad (45)$$

with a Gaussian kernel where the default setting is $c_u = 1$ and where $\hat{h}_{S,opt}$ denotes the plug-in bandwidth (56). In the **Specify Estimation Panel**, see Section 9.1, the user may, however, modify this bandwidth by selecting a different factor c_u .

JMulTi computes $\hat{\mu}_s(\cdot, \hat{h}_{S,u})$, $s = 1, \dots, S$, at

- a user given vector x where each element is by default set to the mean of y_t and can be changed in the **Specify Estimation Panel** in the **Evaluate function at** box.
- on a grid of values such that the function can be plotted, see Section 4.1.3, and the **Specify Estimation Panel**, Section 9.1.

4.1.2 Confidence Intervals

One can also compute confidence intervals if one only allows the function to vary within one direction. They are computed at the grid points based on, $s = 1, \dots, S$,

$$\left[\hat{\mu}_s(x, \hat{h}_{S,u}) - z_{\alpha/2} \sqrt{\frac{\hat{v}_s(x, \hat{h}_{s,B}, \hat{h}_{S,u}, \hat{h}_{s,B})}{T_S \hat{h}_{S,u}^m}}, \quad \hat{\mu}_s(x, \hat{h}_{S,u}) + z_{\alpha/2} \sqrt{\frac{\hat{v}_s(x, \hat{h}_{s,B}, \hat{h}_{S,u}, \hat{h}_{s,B})}{T_S \hat{h}_{S,u}^m}} \right], \quad (46)$$

where the estimator of the asymptotic seasonal estimation variance for season s is given by

$$\widehat{v}_s(x, \widehat{h}_{s,B}, \widehat{h}_{S,u}, \widehat{h}_{s,B}) = \frac{\widehat{\sigma}_s^2(x, \widehat{h}_{s,B}, \widehat{h}_{S,u}) \|K\|_2^{2m}}{\widehat{f}_s(x, \widehat{h}_{s,B})}, \quad s = 1, \dots, S. \quad (47)$$

Here the conditional variance estimator $\widehat{\sigma}_s^2(x, h, \widehat{h}_{S,u})$ (61) and the rule-of-thumb bandwidth (53) are used.

To estimate the seasonal densities $f_s(x)$, $s = 1 \dots, S$, the kernel estimator

$$\widehat{f}_s(x, h) = \frac{1}{T_{M,S}} \sum_{\tau=i_{M,S}}^{T_S} K_h(x_{s+\tau S} - x), \quad s = 1, \dots, S, \quad (48)$$

is used in **JMulTi**. The rule-of-thumb bandwidths $\widehat{h}_{s,B}$, $s = 1, \dots, S$, are given in (53).

In addition to the confidence intervals (46) based on the estimated conditional volatility function $\widehat{\sigma}_s^2(x)$, **JMulTi** also computes confidence intervals based on the assumption of homoskedastic errors by replacing in (47) the conditional variance estimator by the variance estimator (58) below. See Section 9.1 on how to choose the direction of interest i and conditioning values, the significance level α , and the Bonferroni option.

4.1.3 Plotting the seasonal functions

If $m \leq 2$, **JMulTi** estimates the conditional mean functions on a grid of points $[x^{(1)}, \dots, x^{(J)}]$. If x is scalar, the J grid points are equidistantly chosen on the interval $[\min_t y_t, \max_t y_t]$. For a two-dimensional regression vector x , J^2 grid points are used with J equidistant points in direction i covering the interval $[\min_t x_{ti}, \max_t x_{ti}]$, $i = 1, 2$. In the **Specify Estimation Panel**, see Section 9.1, the user can change the default value $J = 5$ in the **Grid points in each direction** box. Since this grid is rectangular, the functions $\mu_s(\cdot)$, $s = 1, \dots, S$, are also computed at points which are outside the data range. All function estimates at points outside the data range are suppressed from the plot since those estimates may exhibit a quite large estimation variance.

Occasionally, the plots are strongly influenced by a few estimates close to the boundary of the sample. If this is the case, then such data points can be expected to have a small density just like points that are completely outside the data range. One therefore may wish to exclude such points from the plots as well. Therefore, one can remove in **JMulTi** from the plots all function estimates at points for which the estimated density $\widehat{f}_s(x, \widehat{h}_{s,B})$ given by (48) is in the b -percent quantile of lowest density values where $\widehat{h}_{s,B}$ is given by (53). This is done in the same panel in the box **Quantile of grid points to be suppressed in plot**.

If there are more than two lags, $m > 2$, and one wants to plot the functions, one lets x vary in the directions of two lags and conditions on the values of the other lags. In the **Specify Estimation Panel** the user can specify in the boxes **1st lag to plot** and **2nd lag to plot** for which lags to plot the function. The conditioning values are by default set to the mean

of y_t and can be changed in the box **Condition plot at**. When interpreting such plots one has to bear in mind that changing the conditioning values also changes the plot.

4.2 Lag and Bandwidth Selection

4.2.1 The formulas

For general information see Section 3.2 on the NAR model. This sections only contains the formulas for the SNAR model that differ from the NAR model.

The Asymptotic Final Prediction Error for the SNAR model is given by

$$AFPE_S(h, i_1, \dots, i_m) = A_S + b_S(h)B_S + c(h)C_S \quad (49)$$

where the integrated variance A_S , the integrated variance of estimation $b_S(h)B_S$ and the integrated squared bias of estimation $c_S(h)C_S$ with the (unknown) constants are given by

$$\begin{aligned} A_S &= \frac{1}{S} \sum_{s=1}^S \int \sigma_s^2(x) w(x_M) f_s(x_M) dx_M \\ B_S &= \frac{1}{S} \sum_{s=1}^S \int \sigma_s^2(x) w(x_M) f_s(x_M) / f_s(x) dx_M, \\ C_S &= \frac{1}{S} \sum_{s=1}^S \int \left(\text{tr} \left\{ \frac{\partial^2 \mu_s(x)}{\partial x \partial x'} \right\} \right)^2 w(x_M) f_s(x_M) dx_M \end{aligned}$$

and

$$b_S(h) = \|K\|_2^{2m} (T_{M,S})^{-1} h^{-m}, \quad c(h) = \sigma_K^4 h^4 / 4. \quad (50)$$

For a given lag combination $\{i_1, \dots, i_m\}$ the $AFPE_S(h, i_1, \dots, i_m)$ can also be used to obtain the asymptotically optimal (global) bandwidth. It is given by

$$h_{S,opt} = \left\{ \frac{m \|K\|_2^{2m} B_S}{T_{M,S} \sigma_K^4 C_S} \right\}^{1/(m+4)}. \quad (51)$$

The following estimators are used: The constant B_S is estimated by

$$\widehat{B}_S(\widehat{h}_{S,B}) = \frac{1}{S} \sum_{s=1}^S \frac{1}{T_{M,S}} \sum_{\tau=i_{M,S}}^{T_S} \left\{ y_{s+\tau S} - \widehat{\mu}(x_{s+\tau S}, \widehat{h}_{s,B}) \right\}^2 \frac{w(x_{s+\tau S, M})}{\widehat{f}_s(x_{s+\tau S}, \widehat{h}_{s,B})} \quad (52)$$

in which $h_{S,B} = \left(h_{1,B} \quad h_{2,B} \quad \dots \quad h_{S,B} \right)'$ and where $\widehat{f}_s(\cdot)$ is the Gaussian kernel estimator (48) of the density $f_s(x)$ and \widehat{h}_B is Silverman's (1986) rule-of-thumb bandwidth

$$\hat{h}_{s,B} = \hat{\sigma}_s \left(\frac{4}{m+2} \right)^{1/(m+4)} T_{M,S}^{-1/(m+4)} \quad (53)$$

with $\hat{\sigma}_s = \left(\prod_{j=1}^m \sqrt{\text{Var}(\mathbf{y}_{s-i_j})} \right)^{1/m}$ denoting the geometric mean of the seasonal standard deviation of the regressors and where $\mathbf{y}_{s-i_j} = \left(y_{s+\tau i_{M,S}-i_j} \quad y_{s+\tau(i_{M,S}+1)-i_j} \quad \cdots \quad y_{s+\tau T_S-i_j} \right)'$. The constant C_S is estimated by

$$\hat{C}_S(\hat{h}_{S,C}) = \frac{1}{S} \sum_{s=1}^S \frac{1}{T_{M,S}} \sum_{\tau=i_{M,S}}^{T_S} \left[\sum_{j=1}^m \hat{\mu}_s^{(jj)}(x_{s+\tau S}, \hat{h}_{S,C}) \right]^2 w(x_{s+\tau S, M}), \quad (54)$$

in which $h_{S,C} = \left(h_{1,C} \quad h_{2,C} \quad \dots \quad h_{S,C} \right)'$ and where $\mu_s^{(jj)}(\cdot)$ denotes the second-order direct derivative of the function $\mu_s(\cdot)$. The second-order direct derivatives are estimated with the direct local quadratic estimator (22) that is adapted for each season. The rule-of-thumb bandwidth is given by

$$\hat{h}_{s,C} = 3\hat{\sigma}_s \left(\frac{4}{m+4} \right)^{1/(m+6)} T_{M,S}^{-1/(m+6)}, \quad s = 1, \dots, S. \quad (55)$$

Based on the estimated B_S and C_S , the plug-in bandwidth is given by

$$\hat{h}_{S,opt} = \left\{ \frac{m \|K\|_2^{2m} \hat{B}_S(\hat{h}_{S,B})}{T_{M,S} \hat{C}_S(\hat{h}_{S,C}) \sigma_K^4} \right\}^{1/(m+4)}. \quad (56)$$

The constant A_S is estimated by

$$\hat{A}_S(h) = \frac{1}{S} \sum_{s=1}^S \frac{1}{T_{M,S}} \sum_{\tau=i_{M,S}}^{T_S} \{y_{s+\tau S} - \hat{\mu}_s(x_{s+\tau S}, h)\}^2 w(x_{s+\tau S, M}) \quad (57)$$

In order to eliminate the asymptotic first-order bias from $\hat{A}_S(h)$, the asymptotic bias is subtracted. Thus, the integrated variance is estimated by

$$\hat{A}_S(\hat{h}_{S,u}) - \left\{ b_S(\hat{h}_{S,u}) - \frac{2K(0)^m}{T_{M,S} \hat{h}_{S,u}^m} \right\} \hat{B}_S(\hat{h}_{S,B}) - c(\hat{h}_{S,u}) \hat{C}_S(\hat{h}_{S,C}) \quad (58)$$

where as before $\hat{h}_{S,u} = c_u \hat{h}_{S,opt}$.

Taking all estimates together, one can estimate the $AFPE_S(\cdot)$. Note that **JMULTi** here always uses the plug-in bandwidth $\hat{h}_{S,opt}$. The AFPE is calculated as

$$AFPE_S = \widehat{A}_S(\widehat{h}_{S,opt}) + \frac{2K(0)^m}{T_{M,S}\widehat{h}_{S,opt}^m} \widehat{B}_S(\widehat{h}_{S,B}). \quad (59)$$

Yang and Tschernig (2002) have shown that conducting lag selection based on $AFPE$ is consistent if the underlying data generating process is not linear, that is if $C_S > 0$.

Despite its consistency Tschernig and Yang (2000) have shown that the $AFPE$ tends to include lags in addition to the correct ones. In other words, overfitting is quite likely. Therefore they suggest a corrected Asymptotic Final Prediction Error (CAFPE). The CAFPE is calculated as

$$CAFPE_S = AFPE_S \{1 + m(T - i_m)^{-4/(m+4)}\}. \quad (60)$$

It is also consistent but exhibits much better finite sample behavior. It should therefore be used in practice.

4.2.2 Implementation

In all computations the weight function $w(\cdot)$ is taken to be the indicator function on the data.

Computing \widehat{B}_S : For estimating the density the leave-one-out version of (48) is used. For computing $\widehat{\mu}_S(x_t, \widehat{h}_B)$ the leave-one-out estimator is used as well.

Since function estimates for which the estimated seasonal density is within the 5/S%-quantile of the lowest density values for a given season are likely to be unreliable, they are not used for computing B_S . The number of observations in the denominator is adjusted accordingly. In practice it may happen that the inversion in (44) for a given season fails due to numerical inaccuracies, e.g. if there are only few observations or the bandwidth is relatively small. In this case, `JMulTi` multiplies the current bandwidth by 1.05 and computes (44) with the larger bandwidth. If the inversion still does not work, this procedure is repeated. If no success is reached after 30 iterations, the program stops.

Computing $\widehat{C}_S(\cdot)$: the last two measures are applied as well where (44) is replaced by (22) which is used for each season separately.

4.3 Conditional Volatility Analysis

The seasonal conditional volatility function $\sigma_s(\cdot)$ has to be estimated for the computation of the confidence intervals (46). This estimator will be described next.

4.3.1 Estimation of the Seasonal Volatility Function

For estimating the seasonal volatility function $\sigma_s(x)$ one applies the estimation method for $\sigma(x)$ described in Section 3.3 just for the observations of season s . In contrast to the NAR

model, JMULTi uses for the conditional mean and conditional variance estimation the same lag vector x with lags i_1, \dots, i_m . Also, there is no computation of the plug-in bandwidth for the conditional variance estimator. Instead the rule-of-thumb bandwidth $h_{s,B}$, given by (53), is used. The resulting estimator for the conditional variance function is then

$$\hat{\sigma}_s^2(x, \hat{h}_{s,B}, \hat{h}_{S,u}) = e' \left\{ \mathbf{Z}'_s(x) W_s(x, \hat{h}_{s,B}) \mathbf{Z}_s(x) \right\}^{-1} \mathbf{Z}'_s(x) W_s(x, \hat{h}_{s,B}) \hat{\varepsilon}_s^2(\hat{h}_{S,u}), \quad s = 1, \dots, S, \quad (61)$$

with the seasonal residual vector $\hat{\varepsilon}_s^2(\hat{h}_{S,u}) = \left(\hat{\varepsilon}_{s+i_{M,S}S}^2(\hat{h}_{S,u}), \dots, \hat{\varepsilon}_{s+T_S S}^2(\hat{h}_{S,u}) \right)'$ where each element is obtained with the estimator (44)

$$\hat{\varepsilon}_{s+\tau S}(\hat{h}_{S,u}) = y_{s+\tau S} - \hat{\mu}_s(x_{s+\tau S}, \hat{h}_{S,u}), \quad \tau = i_{M,S}, \dots, T_S, \quad s = 1, \dots, S. \quad (62)$$

Note that the estimator $\hat{\sigma}_s^2(x, \hat{h}_{s,B}, \hat{h}_{S,u})$ can lead to negative estimates. If such a situation arises, JMULTi replaces the local linear estimator (61) by a local constant (Nadaraya-Watson) estimator

$$\hat{\sigma}_{s,NW}^2(x, \hat{h}_{s,B}, \hat{h}_{S,u}) = \frac{\sum_{\tau=s+i_{M,S}S}^{T_S} K_{\hat{h}_{s,B}}(x_{s+\tau S} - x) \hat{\varepsilon}_t^2(\hat{h}_{S,u})}{\sum_{\tau=s+i_{M,S}S}^{T_S} K_{\hat{h}_{s,B}}(x_{s+\tau S} - x)}. \quad (63)$$

4.3.2 Bandwidth and Lag Selection

Since for the seasonal conditional volatility function no plug-in bandwidth and no lag search are available in JMULTi, the relevant formulas are not described here.

If one assumes that the conditional volatility function is independent of the season, that is $\sigma_s(x) = \sigma(x)$, then all the nonseasonal volatility estimators are available, see Section 3.3, with the only exception that the residuals are obtained from (62) for all $s = 1, \dots, S$.

5 SDNAR Models

This section summarizes the formulas and settings for the nonparametric estimation of the seasonal dummy nonlinear autoregressive (SDNAR) model (3). The estimation and lag selection is facilitated if there is an equal number of observations for each season. One then has to guarantee that one has at least $M \geq i_m$ starting values for each season and estimation has to start in $\tau = i_{M,S}$, where $i_{M,S}$ is the smallest integer equal to or greater than M/S . The largest value of the nonseasonal index τ is given by $T_S = \left\lfloor \frac{T}{S} \right\rfloor - 1$, where $[a]$ truncates the fractional portion of a . The number of observations per season that is available for estimation if the largest lag is M is denoted by $T_{M,S} = T_S - i_{M,S} + 1$.

5.1 Estimation and Confidence Intervals

5.1.1 Estimation when Seasonal Dummies are known

If the seasonal dummies b_s , $s = 2, \dots, S$ were known where $b_1 = 0$ for guaranteeing identification, one can transform the SDNAR model (3) into a variant of the NAR model (1) by defining

$$\tilde{y}_{s+\tau S} = y_{s+\tau S} - b_s, \quad \tau = i_{M,S}, \dots, T_S, \quad s = 1, \dots, S. \quad (64)$$

Then the SDNAR model can be written as

$$\tilde{y}_{s+\tau S} = \mu(x_{s+\tau S}) + \sigma(x_{s+\tau S})\xi_{s+\tau S}$$

or simply as

$$\tilde{y}_t = \mu(x_t) + \sigma(x_t)\xi_t, \quad t = 1 + i_{M,S}S, \dots, S + T_S S.$$

It is therefore possible to use all the nonparametric tools described in Section 3.1 for fitting NAR models. One just has to replace in the local linear estimator (10) and in the estimation of confidence intervals (12) y_t by \tilde{y}_t , i_m by $i_{M,S}S$, T by $S + T_S S$. For estimating the variance of the nonparametric estimator that is needed to compute confidence intervals, as described in Section 3.1.3 for NAR models, one uses the standard kernel density estimator

$$\hat{f}_D(x, \hat{h}_B) = \frac{1}{T_{M,S}S} \sum_{t=i_{M,S}S+1}^{S+T_S S} K_{\hat{h}_B}(x_t - x) \quad (65)$$

for estimating the average density $f_D(x) = 1/S \sum_{s=1}^S f_s(x)$.

5.1.2 Estimation of the Seasonal Dummies

In practice, the seasonal dummies b_s , $s = 2, \dots, S$, are unknown and have to be estimated. This can be done by using extending the local linear estimator (10) by seasonal dummy variables. Define the dummy variable $D_{s+\tau S, s'}$ that is 1 if $s = s'$ and 0 otherwise. One then can rewrite the SDNAR model (3) as

$$y_t = \mu(x_t) + \sum_{s'=2}^S b_{s'} D_{t, s'} + \sigma(x_t)\xi_t, \quad t = 1 + i_{M,S}S, \dots, S + T_S S.$$

In order to estimate the seasonal dummies at a given point of interest x one simply includes the dummy variables in the “regressor matrix”

$$\mathbf{Z}_D(x) = \begin{pmatrix} 1 & \cdots & 1 \\ D_{1+i_{M,S}S,2} & \cdots & D_{S+T_S S,2} \\ \vdots & \ddots & \vdots \\ D_{1+i_{M,S}S,S} & \cdots & D_{S+T_S S,S} \\ x_{1+i_{M,S}S} - x & \cdots & x_{S+T_S S} - x \end{pmatrix}'$$

and uses the local linear estimator

$$\widehat{b}_s(x, \widehat{h}_b) = e'_s \left\{ \mathbf{Z}'_D(x) W_D(x, \widehat{h}_b) \mathbf{Z}_D(x) \right\}^{-1} \mathbf{Z}'_D(x) W_D(x, \widehat{h}_b) \mathbf{y}_D \quad (66)$$

where e_s denotes an $m+S$ vector whose s th element is 1 and all other elements 0, $s = 2, \dots, S$, and where

$$W_D(x, h) = \text{diag} \{ K_h(x_t - x) / T_{M,S} \}_{t=1+i_{M,S}S}^T, \quad \mathbf{y}_D = \begin{pmatrix} y_{1+i_{M,S}S} & \cdots & y_T \end{pmatrix}'.$$

For robustification the leave-one-out version of (66) is used. Note that $\widehat{b}_s(x, h)$ varies with x although for the underlying model b_s is assumed constant. One therefore can average across these estimators using a weighted average

$$\bar{b}_s(\widehat{h}_b) = \frac{\sum_{\tau=i_{M,S}}^{T_S} w(x_{s+\tau S, M}) \widehat{b}_s(x_{s+\tau S}, \widehat{h}_b)}{\sum_{\tau=i_{M,S}}^{T_S} w(x_{s+\tau S, M})}, \quad s = 2, \dots, S \quad (67)$$

where $w(\cdot)$ is the same weight function as in Section 3.2.1. Yang and Tschernig (2002) have shown that (67) is consistent if the bandwidth decreases with T as for example in (20). However, there does not exist an optimal asymptotic tradeoff due to the averaging. For this reason the rule-of-thumb bandwidth (20) is used here, $\widehat{h}_b = \widehat{h}_B$.

One now can estimate the conditional mean function $\mu(x)$ and the conditional volatility function $\sigma(x)$ using the “estimated” dependent variables

$$\widehat{y}_{s+\tau S}(\widehat{h}_B) = y_{s+\tau S} - \bar{b}_s(\widehat{h}_B), \quad \tau = i_{M,S}, \dots, T_S, \quad s = 1, \dots, S \quad (68)$$

instead of the ones given in (64) as described in Section 5.1.1.

5.2 Lag and Bandwidth Selection

Using the same arguments as in Section 5.1.1 one can use the lag and bandwidth selection techniques outlines in Section 3.2 once one has replaced y_t by either \widehat{y}_t given by (64) or $\widehat{\widehat{y}}_t$

given by (68), i_m by $i_{M,S}S$, and T by $S + T_S S$. Furthermore, the kernel density estimator (65) is used to estimate the average density $f_D(x) = 1/S \sum_{s=1}^S f_s(x)$ and for estimating the constant C given by (16) the rule-of-thumb bandwidth

$$\hat{h}_C = 3\hat{\sigma} \left(\frac{4}{m+4} \right)^{1/(m+6)} T^{-1/(m+6)} \quad (69)$$

is used. Yang and Tschernig (2002) have shown that this lag selection method is consistent under standard regularity conditions extended to seasonal stochastic processes.

6 SHNAR Models

This section summarizes the formulas and settings for the nonparametric estimation of the seasonal shift nonlinear autoregressive (SHNAR) model (4). The estimation and lag selection is facilitated if there is an equal number of observations for each season. One then has to guarantee that one has at least $M \geq i_m$ starting values for each season and estimation has to start in $\tau = i_{M,S}$, where $i_{M,S}$ is the smallest integer equal to or greater than M/S . The largest value of the nonseasonal index τ is given by $T_S = \lfloor \frac{T}{S} \rfloor - 1$, where $\lfloor a \rfloor$ truncates the fractional portion of a . The number of observations per season that is available for estimation if the largest lag is M is denoted by $T_{M,S} = T_S - i_{M,S} + 1$.

If the seasonal mean shifts $\delta_2, \dots, \delta_S$ are known, $\delta_1 = 0$ by definition, one can obtain the nonseasonal process

$$z_{s+\tau S} = y_{s+\tau S} - \delta_s, \quad \tau = 0, 1, \dots, T_S, \quad s = 1, \dots, S, \quad (70)$$

that is a NAR process (1) and use all the techniques from Section 3.

If the seasonal mean shifts $\delta_2, \dots, \delta_S$ are unknown, they can be estimated by taking the seasonal averages

$$\hat{\delta}_s = \frac{1}{T_{M,S}} \sum_{\tau=i_{M,S}}^{T_S} (y_{s+\tau S} - y_{1+\tau S}), \quad s = 2, \dots, S \quad (71)$$

This estimator has \sqrt{T} convergence (Yang and Tschernig (2002)) and converges thus faster than the nonparametric estimators. One therefore can replace the unknown seasonal mean shifts by their estimates and continue the analysis of the resulting NAR model for the

$$\hat{z}_{s+\tau S} = y_{s+\tau S} - \hat{\delta}_s, \quad \tau = 0, 1, \dots, T_S, \quad s = 1, \dots, S, \quad (72)$$

as if the δ_s were known.

For estimating the variance of the nonparametric estimator that is needed to compute confidence intervals, as described in Section 3.1.3 for NAR models, one uses the standard kernel density estimator (65) on the standard process z_t for estimating $f(z_t)$, $t = 1 + i_{M,S}S, \dots, S + T_S S$.

7 Linear Models

In the nonparametric module linear models are included for allowing quick comparisons of linear with nonlinear modelling. See Section on **Model Specification** within **VAR Modelling** for details on the lag selection criteria: FPE, AIC, HQ and SC. For each estimated model, JMulTi delivers all lag selection criteria and the estimated white noise variance for each model considered. Except for the SDAR model (7) the parameters of the estimated model are also reported.

For linear models there is no further estimation, model checking or volatility analysis possible within the **Nonparametric Time Series** module. For the AR model (5) and the SDAR model (7) this can be done within the module **VAR analysis**.

It is, however, possible to compute one-step ahead forecasts and conduct rolling over predictions, see Section 12.

8 Model and Lag Selection — The Select Model Panel

After having appropriately transformed the time series, one calls the nonparametric time series module which opens with the **Specify Lag Selection** panel.

8.1 Specify Lag Selection Panel

The panel, see Figure 1, consists of two parts:

1. Model Choice and Data Preparation

In this part one chooses one of the models listed in Section 1 and decides on standardizing the time series before the analysis. This can be helpful if the available data are very small in magnitude in order to avoid numerical problems and bad numbering of the plot axes.

2. Lag Selection

In the lower part one selects the statistical procedure for lag selection.

8.1.1 Model Choice and Data Preparation

Select Model

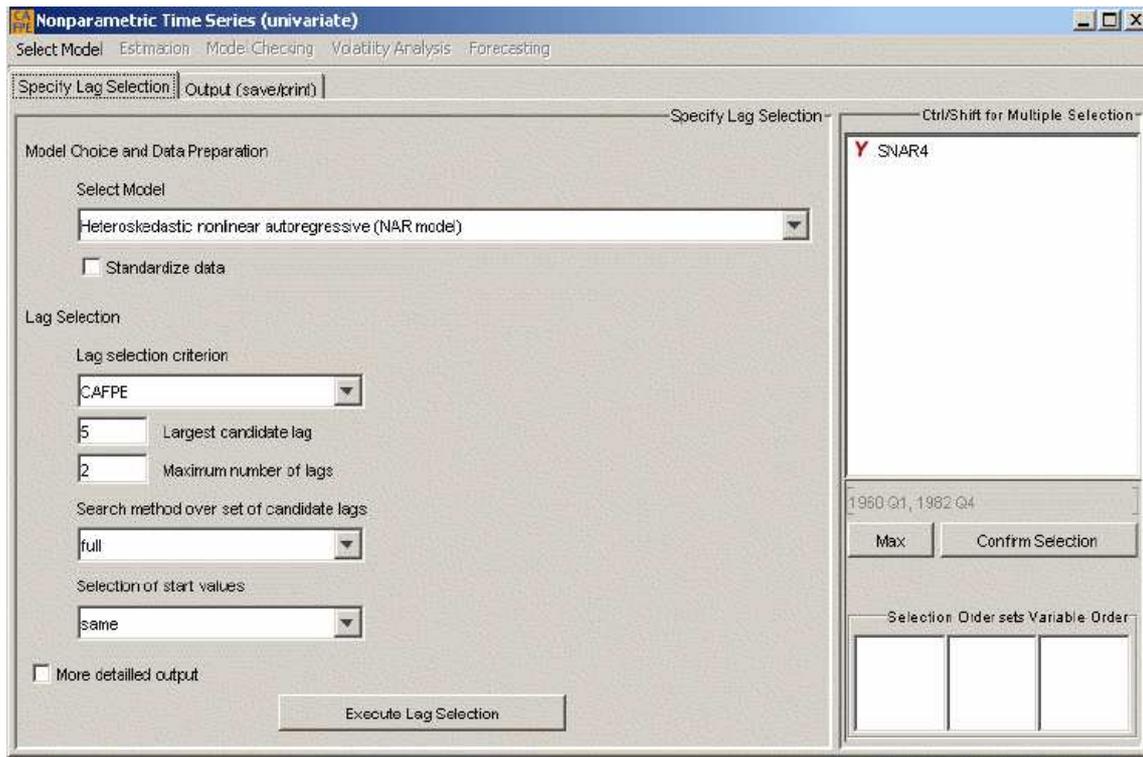


Figure 1: The Lag Selection Panel

First the user chooses a model class listed in Section 1. For yearly data no seasonal models are available. Then one has the choice between

- the Heteroskedastic nonlinear autoregressive (NAR) model (1),
- the Linear autoregressive (AR) model (5).

For quarterly data one additionally obtains the various seasonal variants of these models. For the nonlinear model these are:

- the Heteroskedastic nonlinear autoregressive (NAR) model (1),
- the Seasonal nonlinear autoregressive (SNAR) model (2),
- the Seasonal dummy nonlinear autoregressive (SDNAR) model (3),
- the Seasonal shift nonlinear autoregressive (SHNAR) model (4),

of which the last three allow for various degree of seasonality and for the linear models one has:

- the Linear autoregressive (AR) model (5),
- the Periodic autoregressive (PAR) model (6),

- the Seasonal dummy linear autoregressive (SDAR) model (7),
- the Seasonal shift linear autoregressive (SHAR) model (8).

For the linear models one can conduct lag selection which includes the estimation of the model parameters and do rolling-over out-of-sample forecasts. For any further analysis of linear models, the user has to use the **VAR Analysis** module in JMulTi .

For monthly data, these lists are the same except that the SDNAR and SDAR models are not available.

Standardize data

If the numbers in the time series are either very small or very large, it is recommended to **standardize the data** in order to avoid numerical inaccuracies.

8.1.2 Lag Selection

Lag selection criterion

In practice the relevant set of lags $\{i_1, \dots, i_m\}$ is unknown and has to be selected using a statistical procedure. JMulTi provides several lag selection methods depending on the model class chosen. If a nonlinear model was chosen, then JMulTi offers two nonparametric **lag selection criteria**:

- CAFPE,
- AFPE.

These names indicate the underlying lag selection criterion: the estimated Asymptotic Final Prediction Error (AFPE) or the estimated Corrected Final Prediction Error (CAFPE) are given for the NAR model by (27) and (28), respectively. For the SNAR model the AFPE and CAFPE are calculated by (59) and (60), respectively. For the SDNAR and SHNAR model see Sections 5.2 and 6, respectively.

If a linear model is chosen, then the user can choose among the four standard **lag selection criteria**:

- FPE,
- SC,
- HQ,
- AIC,

where these abbreviations refer to the FPE criterion, the Akaike Information Criterion (AIC), the Schwarz criterion (SC) and the Hannan-Quinn criterion (HQ) all stated in the Section **Model Selection for VAR Modelling** .

Largest candidate lag and Maximum number of lags

All methods have in common that the user has to choose a **largest candidate lag** M implying $i_m \leq M$. Secondly, one has to choose the **maximum number of lags** m where $m \leq M$ holds. Thus, for any selected set of lags $\{i_1, \dots, i_m\}$ it holds that

$$\{i_1, \dots, i_m\} \subseteq \{1, 2, \dots, M\}.$$

Note that if the maximal lag M is large but the number of observations T is not sufficiently large, then due to the curse of dimensionality of nonparametric methods one may have to choose the maximal number of lags m smaller than the maximal lag M .

Search method over set of candidate lags

The choice of M and m is also related to the selection of the **search method** through all possible lag combinations. JMulTi offers two options:

- **full**,
- **directed/sequential**.

A **full** search implies that all possible lag combinations are considered. Since the number of possible lag combinations is given by

$$\sum_{j=1}^m \binom{M}{j}, \quad m \leq M,$$

an increase in M and m leads to a sharp increase in the number of possible lag combinations. For example, $M = m = 8$ implies 255 models while $M = m = 10$ already leads to 1023 models. Since in case of a large number of observations nonparametric methods require intensive computing, it may take too long to conduct a **full search** through all possible lag combinations. In this case, one may prefer to conduct a **directed/sequential search**: lags are added as long as they reduce the selection criterion and one adds that lag from the remaining ones which delivers the largest reduction, see Tjøstheim and Auestad (1994) or Section **Subset Model Selection for VAR Modelling** .

Selection of start values

Finally, JMulTi allows to select the **Selection of start values**:

- same,
- different.

If the strategy **same** is chosen, there are always M start values used. Selecting the strategy **different** implies that the starting values for each lag vector differ since $i'_{m'}$ start values are used for the set of lags $\{i', \dots, i'_{m'}\}$. Note that the lag selection procedures which are implemented in the **VAR Analysis** module always use the **same** strategy.

More detailed output

Activating this option prints out more results about the models that are estimated during the lag selection procedure. Therefore, the additional output depends on the model class chosen. For example, for the heteroskedastic nonlinear autoregressive (NAR) model one additionally obtains for each candidate model the estimated criterion value and error variance plus the estimated constants and rule-of-thumb bandwidths used to compute the plug-in bandwidth, see also the descriptions of the output windows below.

8.2 The Output Window

The output differs somewhat with respect to the selected model. In each subsection below, a typical output is given with a reference to the relevant sections and equations in the help file.

8.2.1 NAR model

JMulTi output

Model and model selection parameters:

Selected model: NAR
 Largest lag considered: 5
 Number of maximal lags: 5
 Search method for lags: full
 Startup values for search: same
 Selection criterion: cafpe
 Robustification of density est.: yes
 Estimator: local linear
 Fraction of obs. screened off
 for estimating A: 0
 for estimating B and C: 0.05

Reference in help

Section 1.1, eq. (1)
 Section 3.2.1
 Section 3.2.1
 Section 8.1.2
 Section 8.1.2
 Section 3.2.1, eq. (27), (28)
 Section 3.1.2, eq. (11)
 Section 3.1.1, eq. (10)
 Section 3.2.2

Estimation of rule-of-thumb bandwidths: Section 3.2.1, eq. (20), (23)
 for B and C: 2 times h.S,
 see Section 5 in TY

Lag vector considered: 1

Lag vector considered: 2

⋮

Results:

I. Main results:

Selected lags/indices of variables using
 lqcafpe:

1 2

Associated criterion value: 0.052256700 Section 3.2.1, eq. (27), (28)

Estimated variance of white noise process: Section 3.2.1, eq.
 0.049387968 (26)

Estimated value of A: 0.047384559 Section 3.2.1, eq. (25)

II. Detailed results:

No. of lags	selected vector of lags/indices
1	1 0 0 0 0
2	1 2 0 0 0
3	1 2 3 0 0
4	1 2 3 4 0
5	1 2 3 4 5

Section 3.2.1, Eq. (24), (20), (23)

No. of lags estim.	est. asymp. opt. bandwidth for A, CAFPE	rule-of-thumb bandwidth for B	rule-of-thumb bandwidth for C
1	0.1404944	0.083415982	0.2170488
2	0.1564908	0.096794487	0.23796443

3	0.17992529	0.1086378	0.25639658
4	0.20175129	0.11906889	0.27272341
5	0.21806246	0.12830897	0.28734557

Section 3.2.1, eq. (27) or (28), (26), (25)

No. of lags estim.	crit. value	est. var of WN	est. value of A (if (C)AFPE)
0	0.073117374	0.073117374	0.073117374
1	0.053834969	0.052419338	0.05299321
2	0.0522567	0.047384559	0.049387968
3	0.056398429	0.043986363	0.049380432
4	0.089438726	0.041858601	0.067123068
5	0.12324969	0.038672169	0.082283728

CAPFE program terminated.

If the **More Detailed Output** box is checked, then one obtains for each model estimated during the lag selection process detailed information. Below is given an example and the relevant references to the equations in the help.

JMulTi output

Reference in help

Lag vector considered: 2 4 5

Number of ddummies: 0

Rule-of-thumb bandwidth for estimating density: 0.10880219

Plug-in bandwidth for estimating A: 0.27245748

Rule-of-thumb bandwidth for estimating B: 0.10880219

Rule-of-thumb bandwidth for estimating C: 0.25678456

Estimated B: 3.3279067

Estimated C: 4.2513349

Plug-in bandwidth was modified by user by multiplying it with 1

The user-specified bandwidth is 0.27245748

Estimated criterion value: 0.12341462

Estimated error variance (bias cor.): 0.099678743

not relevant here

Section 3.2.1, eq. (20)

Section 3.2.1, eq. (24)

Section 3.2.1, eq. (20)

Section 3.2.1, eq. (23)

Section 3.2.1, eq. (19)

Section 3.2.1, eq. (21)

Section 3.1.1, eq. (9)

Section 3.2.1, eq. (27) or (28)

Section 3.2.1, eq. (26)

8.2.2 SNAR model

JMulTi output

Reference in help

Model and model selection parameters:

Selected model: SNAR	Section 1.1, eq. (2)
Largest lag considered: 5	Section 4
Number of seasonal periods: 4	Section 4
Seasonal index of first obs.: 1	Section 4
Number of maximal lags: 5	Section 4
Search method for lags: full	Section 8.1.2
Startup values for search: same	Section 8.1.2
Selection criterion: cafpe	Section 4.2, eq. (59) or (60)
Robustification of density est.: no	Section 4.1.2, eq. (48)
Estimator: local linear	Section 4.1.1, eq. (44)
Fraction of obs. screened off	Section 3.2.2
for estimating A: 0	
for estimating B and C: 0.05	
Estimation of rule-of-thumb bandwidths:	Section 4.2.1, eq. (53), (55)
for (seasonal) B and C: 3 times hat sigma_s	
in h_C,s,	
see Section 5 in YT	
Lag vector considered: 1	
Lag vector considered: 2	
:	

Results:

I. Main results:

Selected lags/indices of variables using

lqcafpe:

1 2

Associated criterion value: 1.2409324	Section 4.2.1, eq. (59) or (60)
Estimated variance of white noise process: 1.0391558	Section 4.2.1, eq. (58)
Estimated value of A: 0.87854874	Section 4.2.1, eq. (57)

II. Detailed results:

No. of lags	selected vector of lags/indices
1	3 0 0 0 0
2	1 2 0 0 0
3	1 2 3 0 0
4	1 2 3 4 0
5	1 2 3 4 5

Section 4.2.1, eq. (56), (53), (55)

No. of lags estim.	est. asymp. opt. bandwidth for A, CAFPE	rule-of-thumb bandwidth for B	rule-of-thumb bandwidth for C
1	0.84121441	0.50047599	1.7863156
2	1.0389025	0.55074242	1.9027937
3	1.1856039	0.59498828	2.0042957
4	1.445014	0.63385914	2.0942206
5	1.4966391	0.66799197	2.1741233

Section 4.2.1, eq. (59) or (60), (58), (57)

No. of lags estim.	crit. value	est. var of WN	est. value of A (if (C)AFPE)
0	1.4937676	1.4937676	1.4937676
1	1.4147312	1.2848181	1.3379717
2	1.2409324	0.87854874	1.0391558
3	1.7311953	0.77123293	1.221916
4	3.2313022	0.7529498	1.9192922
5	5.145759	0.62709141	2.6295519

CAPFE program terminated.

If the **More Detailed Output** box is checked, then one obtains for each model estimated during the lag selection process detailed information. Below is given an example and the relevant references to the equations in the help.

JMulTi output

Reference in help

Lag vector considered: 1 3 4 5	
Number of ddummies: 0	not relevant here
Season 1	Section 4.2.1, eq. (53)
Rule-of-thumb bandwidth for estimating density	
0.62079089	
Season 2	
Rule-of-thumb bandwidth for estimating density	
0.62661431	
Season 3	
Rule-of-thumb bandwidth for estimating density	
0.64383228	
Season 4	
Rule-of-thumb bandwidth for estimating density	
0.64422054	
Plug-in bandwidth for estimating A: 1.5748456	Section 4.2.1, eq. (56)
Rule-of-thumb bandwidths for estimating B:	Section 4.2.1, eq. (53)
Season 1 0.62079089	
Season 2 0.62661431	
Season 3 0.64383228	
Season 4 0.64422054	
Rule-of-thumb bandwidths for estimating C:	Section 4.2.1, eq. (55)
Season 1 2.05104409	
Season 2 2.07028423	
Season 3 2.12717102	
Season 4 2.12845382	
Estimated B: 20243.22	Section 4.2.1, eq. (52)
Estimated C: 0.13552392	Section 4.2.1, eq. (54)
Plug-in bandwidth was modified by user	
by multiplying it with 1	Section 4.1.1, eq. (45)
The user-specified bandwidth is 1.5748456	Section 4.1.1, eq. (45)
Estimated criterion value: 3.6210272	Section 4.2.1, eq. (59) or (60)
Estimated error variance (bias cor.):	Section 4.2.1, eq. (58)
2.1696382	

8.2.3 SDNAR model

The output structure for the SDNAR model is the same as for the NAR model. Below some references to the underlying estimation formulas and algorithms are given.

JMulTi output

Reference in help

Model and model selection parameters:

Selected model: SDNAR	Section 1.1, eq. (3)
Largest lag considered: 5	Sections 5 and 3.2.1
Number of seasonal periods: 4	Section 5
Seasonal index of first obs.: 1	Section 5
Number of maximal lags: 5	Section 3.2.1
Search method for lags: full	Section 8.1.2
Startup values for search: same	Section 8.1.2
Selection criterion: cafpe	Sections 5.2 and 3.2.1, eq. (27), (28)
Robustification of density est.: no	Section 5.1.1, eq. (65)
Estimator: local linear	Sec. 5.1.2, eq. (68) and Sec. 3.1.1, eq. (10)
Estimator of seasonal dummies: full dummy estimator	Sec. 5.1.2, eq. (66)
with bandwidth used for B and leave-one-out method	
Fraction of obs. screened off for estimating A: 0	Section 3.2.2
for estimating B and C: 0.05	
Estimation of rule-of-thumb bandwidths:	
for (seasonal) B and C: 3 times hat sigma_s	Sec. 3.2.1, eq. (20) and
in h_C,s,	Sec. 5.2, eq. (69)
see Section 5 in YT	
for density estimation:	Silverman's rule of thumb with the geometric mean of the variance for all lags. The same bandwidth is used for all seasons.

For the relevant references of the

I. Main results:

see the references in Section 8.2.1 and keep in mind the modifications discussed in Section 5.2.

If the **More Detailed Output** box is checked, one obtains in addition to what is shown for the NAR model also estimates of the seasonal dummies b_s , $s = 2, \dots, S$ that are obtained with the weighted average estimator (67).

8.2.4 SHNAR model

The output structure for the SHNAR model is the same as for the NAR model. Below some references to the underlying estimation formulas and algorithms are given.

JMulTi output

Reference in help

Model and model selection parameters:

Selected model: SHNAR	Section 1.1, eq. (4)
Largest lag considered: 5	Sections 6 and 3.2.1
Number of seasonal periods: 4	Section 6
Seasonal index of first obs.: 1	Section 6
Number of maximal lags: 5	Section 3.2.1
Search method for lags: full	Section 8.1.2
Startup values for search: same	Section 8.1.2
Selection criterion: caepe	Sections 6 and 3.2.1, eq. (27), (28)
Robustification of density est.: no	Section 5, eq. (65)
Estimator: local linear	Sec. 6, eq. (71) and Sec. 3.1.1, eq. (10)
Fraction of obs. screened off	Section 3.2.2
for estimating A: 0	
for estimating B and C: 0.05	
Estimation of rule-of-thumb bandwidths:	
for (seasonal) B and C: 3 times hat sigma_s	Sec. 3.2.1,
in h_C,s,	eq. (20) and
see Section 5 in YT	Sec. 5.2, eq. (69)

When interpreting the

I. Main results:

keep in mind that they are obtained for the estimated nonseasonal process \hat{z}_t given by (72) with the techniques of the NAR model. So all relevant references for this part of the output are given in Section 8.2.1.

If the **More Detailed Output** box is checked, one obtains in addition to what is shown for the NAR model also estimates of the seasonal shifts δ_s , $s = 2, \dots, S$ that are obtained with (71). Note that these estimates are independent of the chosen lag vector since the shift estimates are obtained prior to lag selection.

9 Estimation and Confidence Intervals — The Estimation Panel

9.1 Specify Estimation Panel

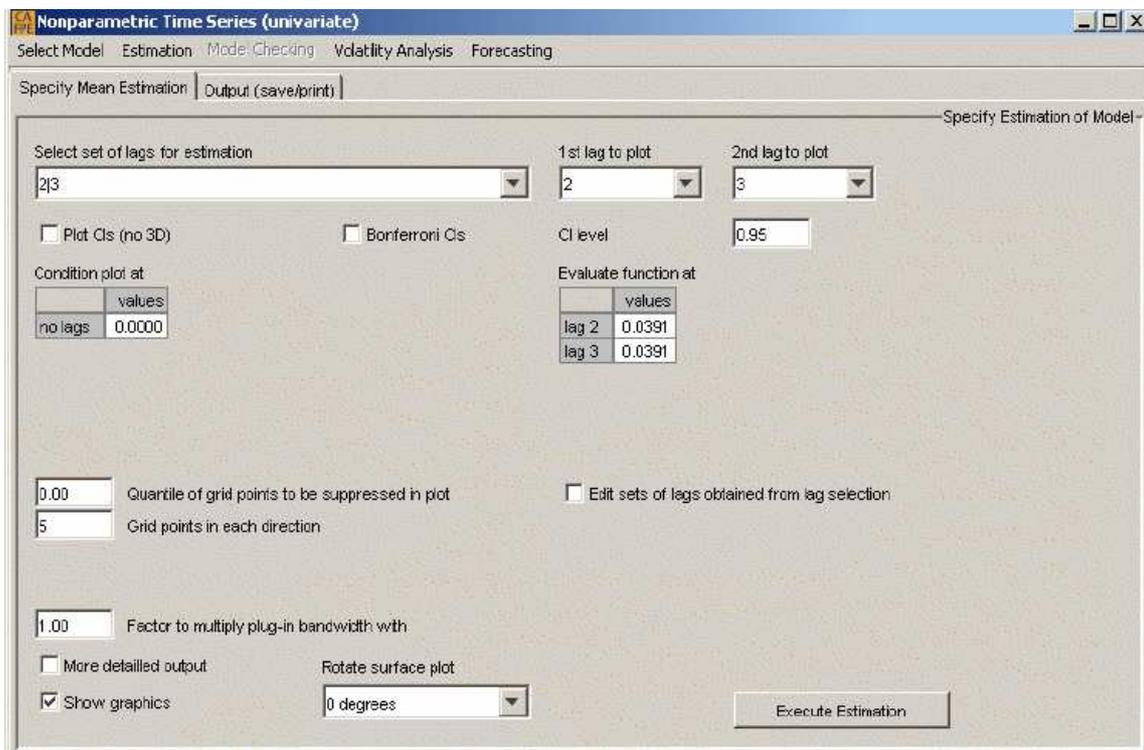


Figure 2: The Specify Mean Estimation Panel

The **Specify Mean Estimation Panel**, see Figure 2, allows to compute and plot nonparametric function estimates. This requires to choose several parameters like the bandwidth, the plotting area, the viewing angle, etc. Section 3 discusses these parameters for the NAR model. Sections 4, 5, and 6 discuss these parameters for the SNAR, SDNAR, and SHNAR models if their definitions differ from those of the NAR model. The following choices have to be made:

Select set of lags for estimation / Edit sets of lags obtained from lag selection

The default lag combination that appears in the **Select set of lags for estimation** box

is the one that exhibits the smallest lag selection criterion among the set of candidate lags. By clicking on the arrow, the optimal lag combinations for the other number of lags m are offered. If for example, the lag selection procedure suggests three lags but the lag selection criterion for two lags is only a small percentage larger, then the user may also investigate the model with two lags.

The user can also change the lags that JMulTi offers for various m by activating the box **Edit sets of lags obtained from lag selection**. This can be useful if one wants to estimate a model for lags obtained from some other lag selection procedure, e.g. a procedure for linear models.

1st lag to plot / 2nd lag to plot / Condition plot at

If the chosen lag vector includes more than two lags, the user can specify for which two lags to plot the function using the **1st lag to plot** and **2nd lag to plot** boxes while keeping the values for all other lags fixed. In JMulTi the default values for the latter ones are the mean of y_t . For each conditional lag the user can change these value in the **Condition plot at** box. Note that by changing the conditioning values, the estimated function plot may change. If $m > 2$, it is recommended to plot the function for several conditioning values.

Evaluate function at

Here the user can specify the point x at which to evaluate the conditional mean function. The estimation result is given in the **Output Panel**. By default each element of x is set to the mean of y_t .

Plot CIs (no 3D) / Bonferroni CIs / CI level

If **Plot CIs (no 3D)** is marked, the estimates of the function including confidence intervals are plotted at each grid point. To allow for a graphical representation the x values can only be varied along one lag while for the other lags the values are conditioned on user specified values that are specified in the box **Condition plot at**. The lag to be varied is chosen in the **1st lag to plot** box. The default confidence level α is 0.95. It can be changed in the **CI level** box. To obtain Bonferroni confidence intervals (14), one has to check the **Bonferroni CIs** box. The confidence intervals based on conditionally heteroskedastic errors are drawn in dashed lines. The confidence intervals based on homoskedastic errors in dotted lines. For the NAR model, the confidence intervals are computed with (12). For the SNAR model, the confidence intervals are based on (46). For the SDNAR and SHNAR model the confidence intervals are given by (12) after the seasonal effects were appropriately removed as described in Sections 5.1 and 6, respectively.

Quantile of grid points to be suppressed in plot

By default the function is plotted only for the range of the data. Since observations which exhibit a small density are prone to a large estimation error, the user can remove the function estimates for which the estimated density at the grid point is smaller than the specified quantile in the box **Quantile of grid points to be suppressed in plot**. Note that the value has to be smaller than 1, see also Section 3.1.2.

Grid points in each direction

Here one chooses the number of grid points J that are used for plotting the estimated mean function in each direction. The default value is 5. For not too large samples $J = 30$ is recommended. For details see Section 3.1.2. Note that the actual number of grid points for which the function is shown can be smaller than J because some grid points are outside the data range or have density estimates below the specified quantile above.

Factor to multiply plug-in bandwidth with

By default, JMulTi uses the plug-in bandwidth of the selected model. However, the user can modify this bandwidth by multiplying the plug-in bandwidth \hat{h}_{opt} by a factor c_u different from the default value 1 as given by (9).

Note that by increasing c_u the plotted function becomes smoother but potentially more biased while by decreasing c_u the plotted function becomes more irregular. Also the probability of numerical failures increases.

This factor does not influence the various rule-of-thumb bandwidths used to obtain the plug-in bandwidth \hat{h}_{opt} . For NAR models the plug-in bandwidth is given by (24). For SNAR models the plug-in bandwidth is given by (56). For SDNAR and SHNAR models the plug-in bandwidth is given by (24) after the seasonal effects were appropriately removed as described in Sections 5.1 and 6, respectively.

More detailed output

By checking this box, one obtains also the rule-of-thumb bandwidths that are used for estimating the density, the constants B and C , the plug-in bandwidth, the user-specified bandwidth, the estimated lag selection criterion, the estimated bias corrected error variance, and in case of seasonal models the estimated seasonal dummies or shifts, respectively. For the relevant formulas for the NAR, SNAR, SDNAR, and SHNAR models see the end of Sections 8.2.1, 8.2.2, 8.2.3, and 8.2.4, respectively.

Show Graphics

By default this box is marked. If one is only interested in computing the function value at specific values, then one should unmark this box to save computation time, especially if the sample and/or the number of lags are large.

Rotate surface plot

Sometimes the function plot has to be rotated in order to be more informative. JMulTi allows choose between four different angles: 0,90, 180, and 270 degrees.

9.2 The Output Window

9.2.1 NAR model

Here only that output is explained that was not explained in Section 8.2.1.

JMulTi output

Model and model estimation parameters:

Selected model: NAR

Lags: 1 2

Number of observations

in time series: 478

for estimation: 476

Number of maximal lags: 2

Multiplication fac. of opt. bandw. 1

Robustification of density est.: yes

Estimator: local linear

Fraction of obs. screened off

for estimating A: 0

for estimating B and C: 0.05

Estimation of rule-of-thumb bandwidths:

for B and C: 2 times h_S, see Section 5 in TY

Estimation:

Plug-in bandwidth was modified by user

by multiplying it with 1

The user-specified bandwidth is 0.1575787

Estimated criterion value: 0.052195486

Estimated error variance (bias cor.):

0.0493691

Variance of dependent variable: 0.072996196

Reference in help

Section 1.1, eq. (1)

Section 3.2.1

Section 3.1.1, eq. (9)

Section 3.1.2, eq. (11)

Section 3.1.1, eq. (10)

Section 3.2.2

Section 3.2.1, eq. (20), (23)

Section 3.1.1, eq. (9)

Section 3.2.1, eq. (27) or (28)

Section 3.2.1, eq. (26)

JMulTi output

The local linear estimates for σ^2 for computing

confidence intervals are negative.

Therefore the Nadaraya-Watson estimator was used

with bandwidth: 0.26108666

Plot settings:

Range of grid for plotted lags 1 :

Minimum: -1.4

Maximum: 1.6

Estimated function values on grid:

Minimum: -1.3126462

Maximum: 1.2792382

Estimated function values within data range:

Minimum: -1.3126462

Maximum: 1.2792382

Function estimates at user-given value:

User-given values of lags:

Lag 1 0.00251046

Lag 2 0.00251046

Estimated function Confidence interval

-0.02666990 [-0.04340, -0.00994]

Estimated conditional error variance

0.04675225

Estimated variance of function estimate

0.00007283

Reference in help

see Section 3.3.1, eq.

(33)

Section 3.1.2: $\min x_{t1}$

$\max x_{t1}$

Section 3.1.1, **Evaluate function at box**

Sections 3.1.1, eq. (10), Section 3.1.3, eq. (12)

Section 3.3.1, eq. (31)

Section 3.1.3, eq. (13)

If the **More Detailed Output** box is checked, one additionally obtains for the chosen model the same information as described at the end of Section 8.2.1 for the case of that box checked plus the largest estimated density value.

9.2.2 SNAR model

Model and model estimation parameters:

Selected model: SNAR
Lags: 1 2
Number of observations
 in time series: 400
 for estimation: 396
Number of seasonal periods: 4
Seasonal index of first obs.: 1
Number of maximal lags: 2

JMulTi output

Multiplication fac. of opt. bandw. 1
Robustification of density est.: no
Estimator: local linear
Fraction of obs. screened off
 for estimating A: 0
 for estimating B and C: 0.05
Estimation of rule-of-thumb bandwidths:
 for (seasonal) B and C: 3 times $\hat{\sigma}_s$
in $h_{C,s}$,
 see Section 5 in YT

Estimation:

Plug-in bandwidth was modified by user by
multiplying it with 1
The user-specified bandwidth is 1.04657
Estimated criterion value: 1.2402579

Estimated error variance (bias cor.):
1.038315
Variance of dependent variable: 1.4920316
The local linear estimates for $\hat{\sigma}^2$ for
computing

Reference in help

Section 4.1.1, eq. (45)
Section 4.1.2, eq. (48)
Section 4.1.1, eq. (44)
Section 3.2.2

Section 4.2.1, eq. (53), (55)

Section
4.1.1, eq. (45)
Section 4.1.1, eq. (45)
Section 4.2.1, eq. (59) or
(60)
Section 4.2.1, eq. (58)

Can happen at some grid
points or

confidence intervals are negative.

Therefore the Nadaraya-Watson estimator was used

with bandwidth: 0.56497067

Plot settings:

Range of grid for plotted lags 1 :

Minimum: -3.9743827

Maximum: 3.3184761

Estimated function values on grid:

Minimum Maximum

Season 1 -0.86278994 0.60236634

Season 2 -1.28220243 0.15912244

Season 3 -0.08657884 1.73539075

Season 4 0.30601623 0.86821286

Estimated function values within data range:

Season 1 -0.86278994 0.55959997

Season 2 -1.26712183 0.15912244

Season 3 -0.08657884 1.43355654

Season 4 0.30601623 0.84617218

JMulTi output

Function estimates at user-given value:

User-given values of lags:

Lag 1 0.03911911

Lag 2 0.03911911

Estimated function Confidence interval

Season 1 -0.14958573 [-0.27489, -0.02428]

Season 2 0.11692264 [-0.04426, 0.27811]

Season 3 0.04401730 [-0.11604, 0.20408]

Season 4 0.35171448 [0.18367, 0.51976]

Estimated conditional error variance

Season 1 0.51532872

Season 2 1.05412327

Season 3 0.93395861

at the point of function evaluation.

See Section 4.3.1, eq. (63)

Section 4.1.3: $\min x_{ti}$

$\max x_{ti}$

Reference in help

Section 4.1.1, **Evaluate function at box**

Sec. 4.1.1, eq. (44), Sec. 4.1.2, eq. (46)

Section 4.3, eq. (61)

Season 4 1.01977286

Estimated variance of function estimate Section 4.1.2, eq. (47)

Season 1 0.00408741

Season 2 0.00676327

Season 3 0.00666900

Season 4 0.00735151

If the **More Detailed Output** box is checked, one additionally obtains for the chosen model the same information as described at the end of Section 8.2.2 for the case of that box checked plus the largest estimated density value.

9.2.3 SDNAR model

JMulTi output

Reference in help

Model and model estimation parameters:

Selected model: SDNAR

Lags: 2 3

Number of observations

 in time series: 400

 for estimation: 396

Number of seasonal periods: 4

Seasonal index of first obs.: 1

Number of maximal lags: 2

Multiplication fac. of opt. bandw. 1

Sections 5.1 and 3.1.1, eq. (9)

Robustification of density est.: no

Section 5.1.1, eq. (65)

Estimator: local linear

Section 5.1 and 3.1.1, eq. (10)

Estimator of seasonal dummies: full dummy estimator

Section 5.1.2, eq. (67)

 with bandwidth used for B

 and leave-one-out method

Fraction of obs. screened off

Section 3.2.2

 for estimating A: 0

 for estimating B and C: 0.05

Estimation of rule-of-thumb bandwidths:

for (seasonal) B and C: 3 times hat sigma_s in h_C,s,	Sec. 3.2.1, eq. (20) and
see Section 5 in YT	Sec. 5.2, eq. (69)
for density estimation: Silverman's rule of thumb	
with the geometric mean of the variance for all lags.	
The same bandwidth is used for all seasons.	
Estimation:	

Plug-in bandwidth was modified by user	Sections 5.1 and 3.1.1, eq. (9)
by multiplying it with 1	
The user-specified bandwidth is 1.122137	Sections 5.1 and 3.1.1, eq. (9)
Estimated criterion value: 1.3426868	Sections 5.2 and 3.2.1, eq. (27), (28)
Estimated error variance (bias cor.): 1.2846364	Sections 5.2 and 3.2.1, eq. (26))
Variance of adjusted dependent variable after subtracting seas. dummies: 1.3977238	Section 5.1.2, estimated var. of \hat{y}_t
The local linear estimates for sigma^2 for computing confidence intervals are negative.	Can happen at some grid points or at the point of function eval- uation.
Therefore the Nadaraya-Watson estimator was used	See Section 3.3.1 eq. (33)
with bandwidth: 0.45066405	
Plot settings:	
Range of grid for plotted lags 2 :	
Minimum: -3.9743827	Section 5.1 and 3.1.2: min x_{ti}
Maximum: 3.3184761	max x_{ti}
Estimated function values on grid:	

Minimum: -0.38675278
 Maximum: 0.35096583
 Estimated function values within data range:
 Minimum: -0.38675278
 Maximum: 0.35096583

JMulTi output

Function estimates at user-given value:

User-given values of lags:

Lag 2 0.03911911

Lag 3 0.03911911

Estimated function Confidence interval

-0.32887876 [-0.40940, -0.24835]

Estimated conditional error variance

1.16748962

Estimated variance of function estimate

0.00168802

For your information:

Estimated parameters of linear AR model

for adjusted data with the same lag vector:

cons -0.36406641

ar2 0.01208118

ar3 -0.01511652

Reference in help

Section 3.1.1, **Evaluate function at box**

Sec. 5.1, 3.1.1, eq. (10), Sec. 3.1.3, eq. (12)

Sections 5.1, 3.3.1, eq. (31)

Sections 5.1, 3.1.3, eq. (13)

If the **More Detailed Output** box is checked, one obtains in addition to what is shown for the NAR model also estimates of the seasonal dummies b_s , $s = 2, \dots, S$ that are obtained with the weighted average estimator (67) plus the largest estimated density value.

9.2.4 SHNAR model

JMulTi output

Reference in help

Model and model estimation parameters:

Plot settings:

Range of grid for plotted lags 1 3 :

Minimum: -4.4925164

Maximum: 2.7988276

Section 6 and 3.1.2: $\min x_{ti}$
 \max_{ti}

Values of conditioning lags:

Lag 4 0.03911911

Section 6 and 3.1.2

Estimated function values on grid:

Minimum: -1.7185371

Maximum: 0.60630696

Estimated function values within data range:

Minimum: -1.7185371

Maximum: 0.60630696

JMulTi output

Function estimates at user-given value:

Reference in help

Section 3.1.1, **Evaluate function at box**

User-given values of lags:

Lag 1 0.03911911

Lag 3 0.03911911

Lag 4 0.03911911

Estimated function Confidence interval

Sec. 6, 3.1.1, eq. (10), Sec. 3.1.3, eq. (12)

-0.41953673 [-0.51951, -0.31957]

Estimated conditional error variance

Sections 6, 3.3.1, eq. (31)

1.45294917

Estimated variance of function estimate

Sections 6, 3.1.3, eq. (13)

0.00260161

For your information: Estimated parameters of linear AR model

for seasonally adjusted data with the same lag vector:

cons -0.43153934

ar1 -0.08482477

ar3 -0.00850772

ar4 0.02273316

If the **More Detailed Output** box is checked, one obtains in addition to what is shown for the NAR model also estimates of the seasonal mean shifts δ_s , $s = 2, \dots, S$ that are obtained with the estimator (71) plus the largest estimated density value.

10 Model Checking — The Model Checking Panel

For model checking one has the same options as for the residuals of VAR models, see Figure 3: a Portmanteau test, an univariate ARCH-LM test, an LM test for autocorrelation (Godfrey test), a test for nonnormality (Jarque-Bera-Lomnicki test). In addition, one can plot or save the residuals, compute their spectrum and estimate their density with a kernel estimator. It has to be noted, however, that the asymptotic distributions of the residual tests are not well analysed in the case of nonparametric residuals. For a detailed description of the various tests see Section **Diagnostic Tests** in the **VAR Chapter** .

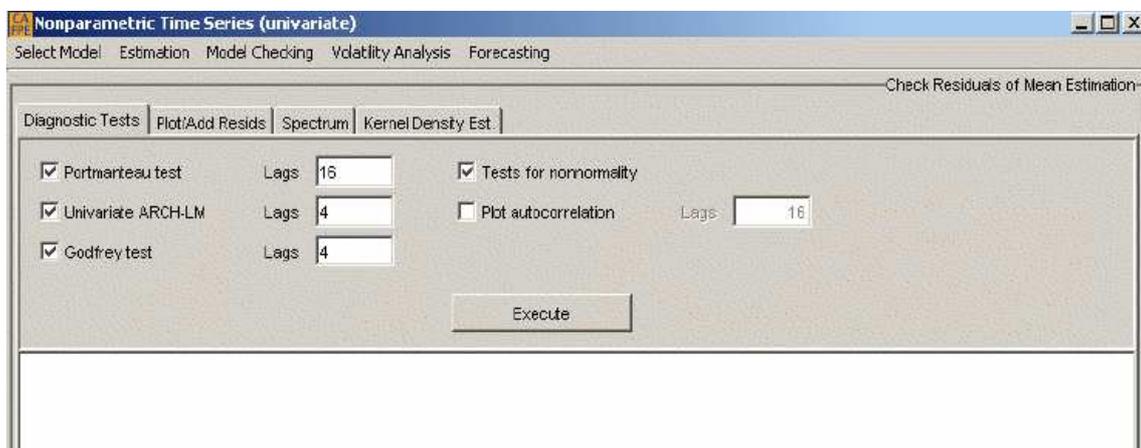


Figure 3: The Model Checking Panel

11 Analysis of Conditional Volatility — The Volatility Analysis Panel

11.1 The Specify Lag Selection for Volatility Panel

The lag selection panel, see Figure 4, looks a bit similar to the one available for the conditional mean function, see Section 8.1. Note, however, that the lag selection depends on the residuals of the estimated conditional mean function. Since the latter also depends on the user-specified bandwidth \hat{h}_u , see Section 9.1, so does the lag selection!

Select Model

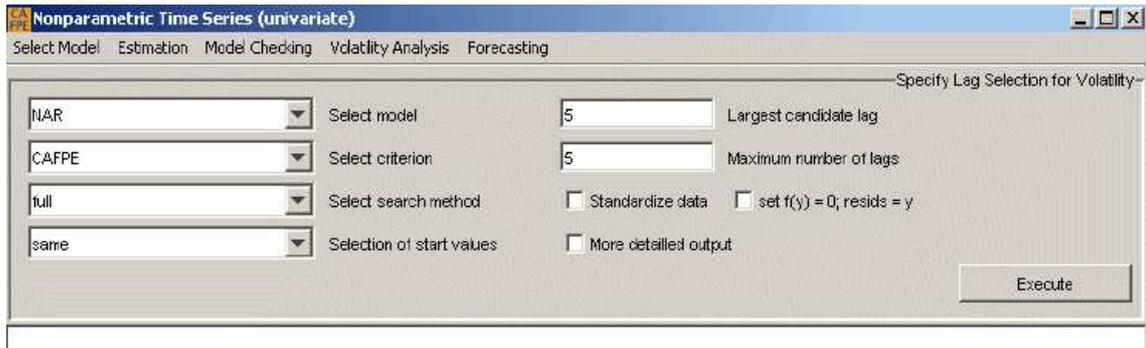


Figure 4: The Specify Lag Selection for Volatility Panel

There are two model classes available

- the modified NAR model (29),
- the modified SDNAR model

The SDNAR model for the conditional volatility function assumes

$$\sigma_s^2(x) = \sigma^2(x) + b_{\sigma,s}, \quad s = 1, \dots, S.$$

It is estimated by adapting the estimation and lag selection techniques developed for the SDNAR model (3) for the conditional mean function, see Section 5, to the process

$$\varepsilon_{s+\tau S}^2 = \sigma^2(x_{s+\tau S}) + b_{\sigma,s} + (\sigma^2(x_{s+\tau S}) + b_{\sigma,s}) (\xi_{s+\tau S}^2 - 1), \quad \tau = i_{M,S}, \dots, T_S, \quad s = 1, \dots, S.$$

Select criterion

JMulTi offers two nonparametric lag selection criteria:

- CAFPE given by (42),
- AFPE given by (41).

Largest candidate lag and Maximum number of lags

One has to choose a **Largest candidate lag** M_σ implying $i_{\sigma,m} \leq M_\sigma$, see Section 3.3.1. Secondly, one has to choose the **Maximum number of lags** m_σ where $m_\sigma \leq M_\sigma$ holds. Thus, for any selected set of lags $\{i_{\sigma,1}, \dots, i_{m_\sigma}\}$ it holds that

$$\{i_{\sigma,1}, \dots, i_{m_\sigma}\} \subseteq \{1, 2, \dots, M_\sigma\}.$$

Since the bandwidth and lag selection procedures for the NAR model (1) assume that the lags of the conditional volatility function are contained in the set of lags of the conditional

mean function, one should set now M_σ to the largest lag of the conditional mean function. JMulTi allows, however, to choose other values as well.

Select search method

The choice of M_σ and m_σ is also related to the selection of the **search method** through all possible lag combinations. JMulTi offers two options:

- full,
- directed/sequential.

See Section 8.1.2 for details.

Selection of start values

Finally, JMulTi allows to select the **strategy for start values**:

- same,
- different.

See Section 8.1.2 for details.

Standardize data

If the numbers in the time series are either very small or very large, it is recommended to standardize the data in order to avoid numerical inaccuracies.

set f(y)=0; resids=y

If this box is checked, it is assumed in the NAR model (1) that $f(\cdot) = 0$. In this case the asymptotic properties of the nonparametric lag selection procedure are known, see Section 3.3.2.

More detailed output

Activating this option prints out more details of the lag selection procedure.

11.1.1 The Output Window

JMulTi output

Model and model selection parameters:

Selected model: NAR.RES

Reference in help

Section 3.3.1, eq. (29)

Largest lag considered: 5	Section 11.1
Dependent var. for vol. est. resid	Section 3.3.1, eq. (30)
Number of maximal lags: 5	Section 11.1
Search method for lags: full	Section 11.1
Startup values for search: same	Section 11.1
Selection criterion: cafpe	Section 3.3.2, eq. (42) or (41)
Robustification of density est.: yes	Section 3.1.2, eq. (11)
Estimator: local linear	Section 3.3.1, eq. (31)
Fraction of obs. screened off	Section 3.2.2
for estimating A: 0	
for estimating B and C: 0.05	
Estimation of rule-of-thumb bandwidths:	Section 3.2.1, eq. (20), (23)
for B and C: 2 times h _S ,	
see Section 5 in TY	
Lag vector considered: 1	

⋮

Results:

I. Main results:

Selected lags/indices of variables using
lqcafpe:

2 3 4

Associated criterion value: 32.186489	Section 3.3.2, eq. (41) or (42)
---------------------------------------	---------------------------------

Estimated variance of white noise process: 28.023850	Section 3.3.2, eq. (40)
---	-------------------------

Estimated value of A: 22.616863	Section 3.3.2, eq. (39)
---------------------------------	-------------------------

II. Detailed results:

No. of lags selected vector of lags/indices

1 5 0 0 0 0

2 4 5 0 0 0

```

3 2 3 4 0 0
4 1 2 3 5 0
5 1 2 3 4 5

```

Section 3.3.2, eq. (38), (20), (23)

No. of lags estim.	est. asymp. opt. bandwidth for A, CAFPE	rule-of-thumb bandwidth for B	rule-of-thumb bandwidth for C
1	0.70580796	0.49545813	1.3468487
2	0.94736621	0.58971019	1.4967764
3	1.2298677	0.67327542	1.6281021
4	1.5688133	0.74745893	1.7451211
5	2.120392	0.81391603	1.8511669

Section 3.3.2, eq. (41) or (42), (40), (39)

No. of lags estim.	crit. value	est. var of WN	est. value of A (if (C)AFPE)
0	46.57824	46.57824	46.57824
1	41.636589	40.95766	41.246061
2	36.167125	32.806977	34.470385
3	32.186489	22.616863	28.02385
4	57.341679	22.587037	43.87247
5	163.93045	20.142672	110.79945

CAPFE program terminated.

If the **More Detailed Output** box is checked, then one obtains for each model estimated during the lag selection process detailed information. Below is given an example and the relevant references to the equations in the help.

JMulTi output

```

Lag vector considered: 1
Number of ddummies: 0
Rule-of-thumb bandwidth for estimating
density: 0.49482842
Plug-in bandwidth for estimating A: 0.8545354
Rule-of-thumb bandwidth for estimating B:
0.49482842
Rule-of-thumb bandwidth for estimating C:
1.3451369

```

Reference in help

```

not relevant here
Section 3.2.1, eq.
(20)
3.3.2, eq. (38)
Section 3.2.1, eq. (20)
Section 3.2.1, eq. (23)

```

Estimated B: 408.32055
Estimated C: 0.24709865

Section 3.3.2, eq. (36))
Section 3.3.2, eq. (37)

Plug-in bandwidth was modified by user
by multiplying it with 1
The user-specified bandwidth is 0.8545354
Estimated criterion value: 45.269155
Estimated error variance (bias cor.):
44.928171

Section 3.3.1, eq. (32)
Section 3.3.1, eq. (32)
Section 3.3.2, eq. (41) or
(42)
Section 3.3.2, eq. (40)

11.2 The Volatility Estimation Panel

The estimation panel, see Figure 5, looks quite similar to the one available for the conditional mean function, see Section 9.1. Note, however, that the estimation depends on the residuals of the estimated conditional mean function except if the box **Set conditional mean to zero ($f(y)=0$; resids= y)** is activated. Since the residuals also depends on the bandwidth \hat{h}_u that the user specified for the conditional mean estimation, so does the volatility estimation!

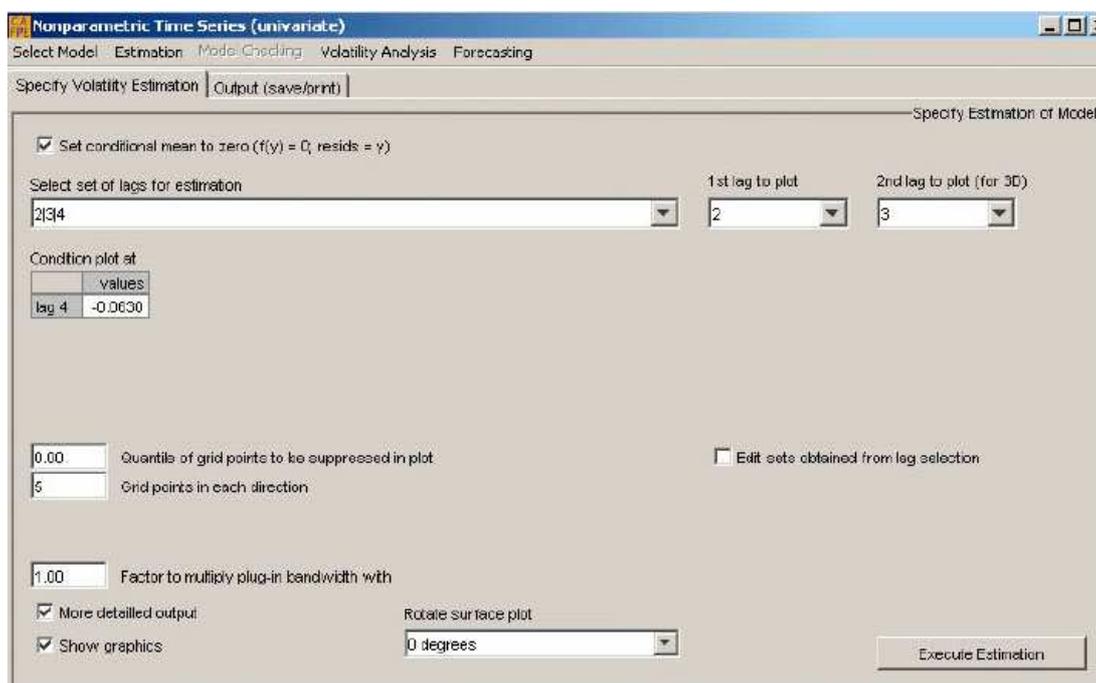


Figure 5: The Specify Volatility Estimation Panel

Set conditional mean to zero ($f(y)=0$; resids= y)

If this box is checked, it is assumed in the NAR model (1) that $f(\cdot) = 0$. In this case the

asymptotic properties of the nonparametric lag selection procedure are known, see Section 3.3.2. This is useful if the conditional mean function is negligible as is sometimes the case for financial data. Moreover, this option can be used to investigate the importance of the estimated conditional mean function on the conditional volatility estimates.

Select set of lags for estimation / Edit sets obtained from lag selection

By default the lag combination with the smallest lag selection criterion is chosen. By clicking on the arrow, the optimal lag combinations for other number of lags are offered. If for example, the lag selection procedure suggests three lags but the lag selection criterion for two lags is only a small percentage larger, then the user may also investigate the model with two lags.

The user can also change the lags that JMulTi offers by activating the box **Edit sets obtained from lag selection** and changing the appropriate lag vectors.

1st lag to plot / 2nd lag to plot(for 3D) / Condition plot at

If the chosen lag vector includes more than two lags, the user can specify for which two lags to plot the function while keeping the values for all other lags fixed. By default, JMulTi sets these values equal to the mean of y_t . For each lag the user can change this value in the **Condition plot at** line.

Quantile of grid points to be suppressed in plot / Grid points in each direction

By default the function is plotted only on the range of the data where the default number of grid points in each direction is 5. Since observations which exhibit a small density are prone to a large estimation error, the user can remove the function estimates for the points which are within the given quantile of the density by choosing in the **Quantile of grid points to be suppressed in plot** box a value between zero and smaller than one.

Factor to multiply plug-in bandwidth with

Here the user can set $c_{\sigma,u}$, see equation (32), to another value than 1. The bandwidth for estimating the conditional mean function is $c_{\sigma,u}\hat{h}_{\sigma,opt}$ where $\hat{h}_{\sigma,opt}$ is given by (38). Note that by increasing $c_{\sigma,u}$ the plotted function becomes smoother but potentially more biased while by decreasing $c_{\sigma,u}$ the plotted function becomes more irregular. Also the probability of numerical failures increases. Remember that the estimation results also depend on the estimated errors and thus on the bandwidth \hat{h}_u selected for the conditional mean function. Also keep in mind that the plug-in bandwidth for the volatility estimation $\hat{h}_{\sigma,opt}$ given by (38) can be misleading if the stochastic process is homoskedastic, see the second last paragraph of Section 3.3.2.

More Detailed Output

By checking this box one obtains results that were used to compute the plug-in bandwidth plus the largest estimated density value.

Show Graphics

By default this box is marked. If one is only interested to compute the function value at specific values, then one should unmark this box to save computation time, especially if the sample and/or the number of lags are large.

Rotate surface plot

Sometimes the function plot has to be rotated in order to be more informative. JMulTi allows to choose between four different angles: 0,90, 180, and 270 degrees.

11.2.1 The Output Window

JMulTi output

Model and model estimation parameters:

Selected model: NAR.RES

Lags: 2 3 4

Number of observations

in time series: 1028

for estimation: 1024

Dependent var. for vol. est. resid

Number of maximal lags: 3

Multiplication fac. of opt. bandw. 1

Robustification of density est.: yes

Estimator: local linear

Fraction of obs. screened off

for estimating A: 0

for estimating B and C: 0.05

Estimation of rule-of-thumb bandwidths:

for B and C: 2 times h.S, see Section 5 in TY

Estimation:

Reference in help

Section 3.3.1, eq. (29)

Section 3.3.1, eq. (30)

Section 3.3.1, eq. (32)

Section 3.1.2, eq. (11)

Section 3.3.1, eq. (31)

Section 3.2.2

Section 3.2.1, eq. (20), (23)

Plug-in bandwidth was modified by user
 by multiplying it with 1 Section 3.3.1, eq. (32)
 The user-specified bandwidth is 1.2296555 Section 3.3.1, eq. (32)
 Estimated criterion value: 32.147393 Section 3.3.2, eq. (41) or (42)
 Estimated error variance (bias cor.): Section 3.3.2, eq. (40)
 27.991689
 Variance of dependent variable: 46.66236

The local linear estimates for σ^2 on the grid are negative.

Therefore the Nadaraya-Watson estimator was Can happen at some grid points.
 used See Section 3.3.1 eq. (33)
 with bandwidth: 1.2296555

Plot settings:

Range of grid for plotted lags 2 3 :

Minimum: -9.5756091 Section 3.1.2: $\min x_{t1}$
 Maximum: 7.5526755 $\max x_{t1}$

Values of conditioning lags:

Lag 4 -0.06302198 Section 11.2, **Condition plot at box**

Estimated function values on grid:

Minimum: 1.2918056
 Maximum: 6.7194156

Estimated function values within data range:

Minimum: 1.4447175
 Maximum: 5.9019027

If the **More Detailed Output** box is checked, one additionally obtains for the chosen model the same information as described at the end of Section 11.1.1 for the case of that box checked plus the largest estimated density value.

11.3 The Check Residuals of Volatility Estimation Panel

In this panel, see Figure 6, one can check the residuals ((43) of the specified and estimated NAR model (1) and see whether they correspond to the assumption of being i.i.d. with mean 0 and variance 1.

To check the residuals the same battery of procedures is available as for VAR models: a Portmanteau test, an univariate ARCH-LM test, and an LM test for autocorrelation (Godfrey

test). In addition, one can plot or save the residuals, compute their spectrum and estimate their density with a kernel estimator. It has to be noted, however, that the asymptotic distributions of the residual tests are not well analysed in the case of nonparametric residuals. For a detailed description of the various tests see Section **Diagnostic Tests** in the **VAR Chapter**.

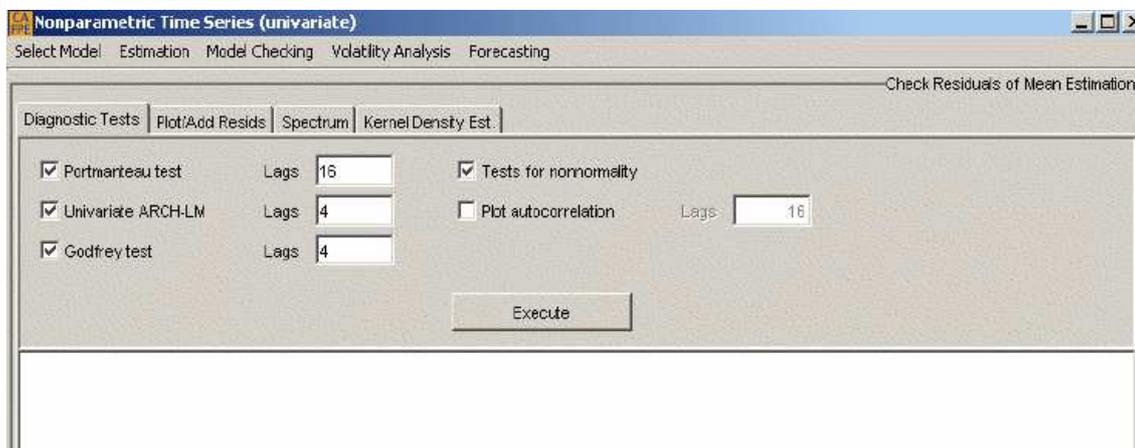


Figure 6: The Check Residuals of Volatility Estimation Panel

12 The Forecasting Panel

12.1 The Specify Forecast Panel

In this panel, see Figure 7, the user can either conduct rolling over, out-of-sample forecasts or one-step ahead forecasts.

Select set of lags for forecasting / Edit sets of lags obtained from lag selection

This works in the same way as in the Estimation Panel, see Section 9.1.

Factor to multiply plug-in bandwidth with

This allows the user to modify the plug-in bandwidth for the one-period ahead forecast or the rolling over, out-of-sample forecasts.

	NAR	SNAR	SDNAR	SHNAR
User-defined bandwidth	(9)	(45)	(9)	(9)

Rolling over, out-of-sample forecasts / One-step ahead forecast

You have either to mark the box **Rolling over, out-of-sample forecasts** or **One-step ahead forecast**

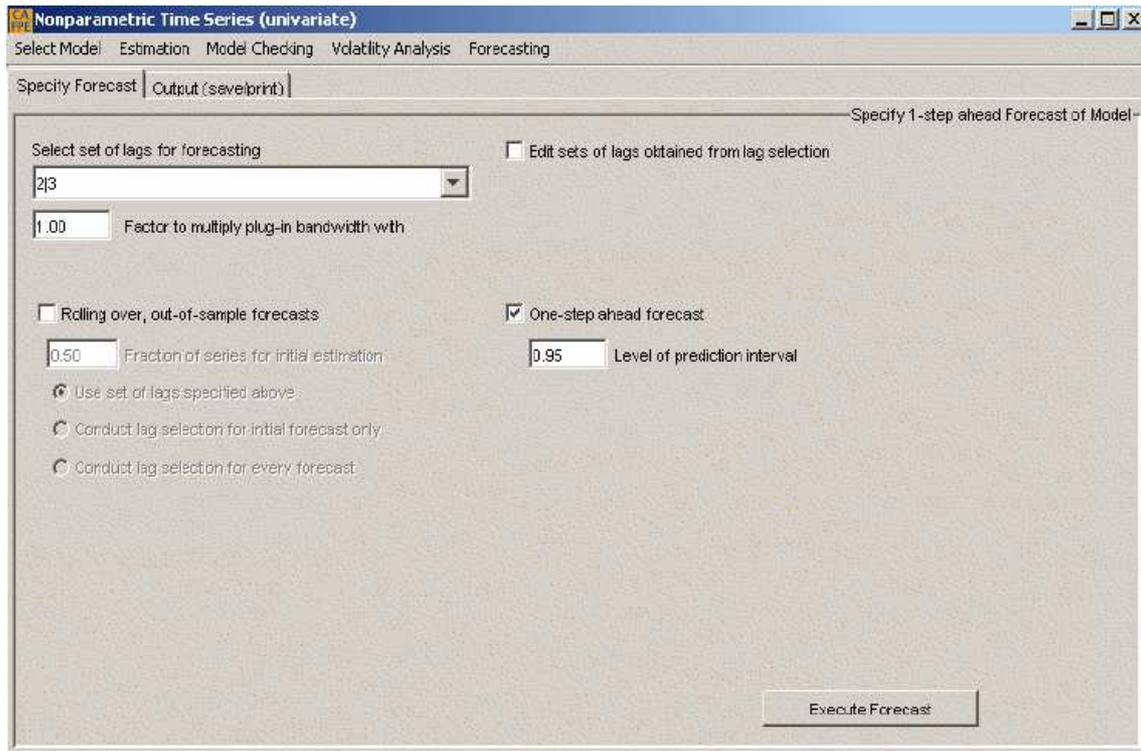


Figure 7: The Specify Forecast Panel

For further explanation of conducting **Rolling over, out-of-sample forecasts**, see Section 12.1.1, for details on an **One-step ahead forecast** see Section 12.1.2.

12.1.1 Rolling-over, out-of-sample forecasts

One splits the data set into a sample for estimation which contains the first T' values and a sample of the remaining $T - T'$ values for out-of-sample forecasting. The first forecast is computed for $y_{T'+1}$ based on all observations available up to time T' . In the next step one forecasts $y_{T'+2}$ based on the sample $\{y_1, \dots, y_{T'+1}\}$. This procedure is iterated until $T - 1$ observations are used and y_T is forecasted. One then computes the rolling over one-step ahead prediction error

$$PE = \frac{1}{T - T'} \sum_{j=T'+1}^T (\hat{y}_j - y_j)^2. \quad (73)$$

In the output this quantity is called MEAN SQUARED PREDICTION ERROR (MSEP).

The use of rolling over predictions is to allow out-of-sample comparisons of the model with other models. For example, one may compare the out-of-sample performance of a NAR model with its linear AR counterpart. Therefore, rolling over predictions can also be conducted for all linear models. Rolling over, out-of-sample predictions can also be computed for all seasonal models: SNAR, SDNAR and SHNAR.

One can also check if the rolling over, out-of-sample forecasts have a smaller variance than the observations to predict, y_j , $j = T' + 1, \dots, T$. At the end of each output one therefore finds the measure

$$\frac{PE - \widehat{Var}(y_j)}{\widehat{Var}(y_j)}$$

Fraction of series for initial estimation

Here one specifies T' in (73) as **Fraction of series ... *T**.

Use set of lags specified above / Conduct lag selection for initial forecast only / Conduct lag selection for every forecast

There are three modes on how to forecast the \hat{y}_j 's in the prediction error (73). In the first mode activated by checking the **Use set of lags specified above** box, all forecasts are computed by using a user-specified lag vector chosen in the **Select set of lags for forecasting** box.

In the second mode, activated by the **Conduct lag selection for initial forecast only**, a lag selection is conducted based on the initial estimation sample up to time T' and then this selected lag vector is maintained for all remaining periods. The lag selection is conducted with the parameters set in the **The Specify Lag Selection Panel**, see Section 8.1.

In the third mode, activated by the **Conduct lag selection for every forecast** box, a lag selection is carried out for each forecast. The lag selection is conducted with the parameters set in the **The Specify Lag Selection Panel**, see Section 8.1. The latter mode can be computational very demanding if the sample size is large and/or a full search is conducted for each lag selection.

12.1.2 One-step ahead forecast

The nonparametric one-period ahead forecast for a time series is obtained by estimating $f(x_{T+1})$ where this estimate is computed using the settings specified in the **The Specify Lag Selection Panel** and the **Specify Estimation Panel**, see Sections 8.1 and 9.1, respectively. For the NAR (1), SDNAR (3), and SHNAR (4) model the prediction interval is based on (12) where $\sqrt{\cdot}$ is replaced by $\sqrt{\widehat{v}(x_{T+1}, \dots) / (T\widehat{h}_h) + \widehat{\sigma}^2(x_{T+1}, \dots)}$. For the SNAR model (2), it works similarly based on (46).

The confidence level is set in the **Level of prediction interval** box.

In practice, one is sometimes interested in h -step ahead forecasts. One can obtain such forecasts if one requires for the first lag $i_1 = h$. Then it is possible to construct $x_{T+h} = (y_{T+h-i_1}, \dots, y_{T+h-i_m})$ where the first component is just y_T .

12.2 The Output Window

The output is hoped to be self-explanatory.

References

- Lütkepohl, H. (1991). *Introduction to multiple time series analysis*, Springer Verlag, Heidelberg.
- Silverman, B. (1986). *Density estimation for Statistics and Data Analysis*, Chapman and Hall, London.
- Tjøstheim, D. and Auestad, B. (1994). Nonparametric identification of nonlinear time-series - selecting significant lags, *Journal of the American Statistical Association* **428**(89). 1410-1419.
- Tschernig, R. (2004). Nonparametric time series modelling, *in* H. Lütkepohl and M. Kräätzig (eds), *Applied Time Series Econometrics*, Cambridge University Press, Cambridge.
- Tschernig, R. and Yang, L. (2000). Nonparametric lag selection for time series, *Journal of Time Series Analysis* **21**(4). 457-487.
- Yang, L. and Tschernig, R. (1999). Multivariate bandwidth selection for local linear regression, *Journal of the Royal Statistical Society, Series B* **61**: 793–815.
- Yang, L. and Tschernig, R. (2002). Non- and semiparametric identification of seasonal nonlinear autoregression models, *Econometric Theory* **18**: 1408–1448.