Discussion Paper No. 13-076

Finding Starting-Values for Maximum Likelihood Estimation of Vector STAR Models

Frauke Schleer

ZEW
Zentrum für Europäische Wirtschaftsforschung GmbH
Centre for European Economic Research

Discussion Paper No. 13-076

Finding Starting-Values for Maximum Likelihood Estimation of Vector STAR Models

Frauke Schleer

Finding Starting-Values for Maximum Likelihood Estimation of Vector STAR Models*

Frauke Schleer†

October 7, 2013

Abstract

This paper focuses on finding starting-values for maximum likelihood estimation of Vector STAR models. Based on a Monte Carlo exercise, different procedures are evaluated. Their performance is assessed w.r.t. model fit and computational effort. I employ i) grid search algorithms, and ii) heuristic optimization procedures, namely, differential evolution, threshold accepting, and simulated annealing. In the equation-by-equation starting-value search approach the procedures achieve equally good results. Unless the errors are cross-correlated, equation-by-equation search followed by a derivative-based algorithm can handle such an optimization problem sufficiently well. This result holds also for higher-dimensional VSTAR models with a slight edge for the heuristic methods. Being faced with more complex Vector STAR models for which a multivariate search approach is required, simulated annealing and differential evolution outperform threshold accepting and the grid with a zoom.

JEL *classification:* C32, C61, C63

Keywords: Vector STAR model, starting-values, optimization heuristics, grid search, estimation, non-linearieties

*Research for this paper has been supported by the SEEK Research Programme of the Centre for European Economic Research (ZEW). For further information on projects of the author see `www.zew.de/staff_fsl` as well as the ZEW annual report on `www.zew.de/en`. I am especially indebted to my SEEK project partners of the project "Financial Stress and Economic Dynamics: Asymmetries Within and Across Euro Area Countries": Timo Teräsvirta, Willi Semmler, and Peter Winker. They had highly valuable comments for this paper. I thank all participants of the SEEK workshop "Non-Linear Economic Modeling: Theory and Applications" for inspiring discussions. I am also grateful to the participants of the ZEW Brown-Bag Seminar and Andreas Sachs. All remaining errors are my own.

†Centre for European Economic Research (ZEW), P.O. Box 103443, D-68034 Mannheim, Germany, Phone: +49/621/1235-164, Fax: +49/621/1235-223, E-mail: schleer@zew.de

1 Introduction

Whatever the use of an econometric model, estimating its parameters as well as possible given the available information is of crucial importance. In the paper I focus on the estimation of multivariate Vector Smooth Transition Autoregressive (VSTAR) models. Thereby, a non-linear optimization function originates which is commonly solved by a numerical, derivative-based algorithm. Yet, parameter estimation may come along with optimization difficulties. Problems due to flat or non-convex likelihood functions may arise.

Hence, in empirical applications derivative-based optimization algorithms may either converge slowly to the global optimum or only to a local optimum which is not globally optimal (Maringer and Winker 2009). As Teräsvirta and Yang (2013) point out the estimation outcome of Vector STAR models crucially relies on good starting-values. Initializing a derivative-based algorithm with starting-values close to the global optimum (or at least a useful local optimum) helps the algorithm to cover the remaining distance to the nearest optimum. Yet, in high-dimensional models trying several starting-values in order to get close the global optimum may turn out to be extremely time-consuming or even not solvable in reasonable computing time. The present paper focuses on finding starting-values for maximum likelihood estimation of Vector STAR models to solve this difficult problem. Different procedures for deriving starting-values are evaluated and their performance is assessed with respect to model fit and computational effort.

Optimization problems of objective functions that are ill-behaved are well-known. One of the pioneering contributions is the work of Goffe, Ferrier, and Rogers (1994) who discuss optimization problems related to the use of conventional optimization techniques by applying optimization heuristics. Typically, regime-switching models are also afflicted with such optimization problems (van Dijk, Teräsvirta, and Franses 2002). Hence, the success of solving a non-linear optimization problem depends on finding suitable starting-values for the subsequent derivative-based optimization procedure.

It is common to apply grid search methods for finding initial values. Yet, it is not clear whether other methods, namely heuristics, could perform better in generating starting-values. The inherent stochastics and potential downhill moves of heuristic optimization procedures may deliver advantages in terms of efficiency and the extent to which the surface area can be explored. This is particularly important when equation-by-equation estimation is not efficient or not feasible at all and a system-wide estimation is necessary. Recent contributions emphasize the need of employing alternative optimization approaches in non-linear models. These include heuristic methods (see, for instance, Wu and Chang (2002); Chan and McAleer (2002); Baragona, Battaglia, and Cucina (2004); Yang, Tian, and Yuan (2007); Maringer and Meyer (2008); Battaglia and Protopapas (2011); El-Shagi (2011); Baragona and Cucina (2013)). The studies concentrate on parameter estimation in univariate or multivariate regime-switching models, whereas the

Vector STAR model and the starting-value search is not addressed. Before applying a heuristic algorithm to the whole modeling cycle of a non-linear Vector STAR model, one might initially focus on improving the starting-value search. Finding reliable initial values is not limited to Vector STAR models, but relevant for a wide range of non-linear economic problems, such as differential equations or Threshold VAR models.
Based on a comprehensive Monte-Carlo set-up, I address the starting-value search for the estimation of Vector STAR models. Up to now, there exists no analysis which tackles the evaluation of starting-values search procedures comprehensively. Different procedures for finding starting-values are evaluated and their performance is assessed with respect to model fit and computational effort based on simulated Data Generating Processes (DGPs) of Vector STAR models. I employ both grid search algorithms and, following the idea of El-Shagi (2011) and Gonzàlez, Rincon, and Rodriquez (2009) three heuristic optimization procedures: differential evolution (DE), threshold accepting (TA) and simulated annealing (SA).
The main results of this study are as follows. For Vector STAR models without cross-correlated error terms equation-by-equation starting-value search procedures are preferable. Although for Vector STAR models which have zero restrictions in the lag structure a multivariate procedure would be efficient, this result still holds. For the equation-by-equation starting-value search approach, all procedures perform equally well with a slight edge for the heuristic methods. However, as soon as the Vector STAR model has cross-correlated error terms, multivariate starting-value search procedures clearly outperform equation-by-equation approaches. The comparison of different algorithms indicates that SA and DE generate the best outcomes. Yet, SA and DE are more time-consuming than the grid with a zoom, which is due to a lower number of likelihood evaluations of the latter.
The paper is organized as follows. Section 2 introduces the Vector STAR model and considers its characteristics. The competing methods for finding starting-values are described in section 3. Section 4 describes the evaluation framework which is basically a comprehensive Monte Carlo experiment. The simulation results are presented in section 5. Finally, section 6 concludes.

2 The Vector STAR model

The logistic Vector STAR model can capture different dynamic properties across regimes (asymmetric behavior), has a straightforward economic interpretation in different regimes

and can handle smooth transitions from one regime into the other. It looks as follows:[1]

$$\mathbf{y_t} = \Big\{ \sum_{i=1}^{m} (\mathbb{G}_{\boldsymbol{t}}^{\boldsymbol{i-1}} - \mathbb{G}_{\boldsymbol{t}}^{\boldsymbol{i}}) \mathbf{F_i'} \Big\} \mathbf{x_t} + \boldsymbol{\varepsilon_t} \tag{1}$$

where $\mathbf{y_t}$ is a $k \times 1$ column vector and $\mathbf{x_t} = (\mathbf{y'_{t-1}}, \ldots, \mathbf{y'_{t-p}}, \mathbf{d'_t})'$, where $\mathbf{d_t}$ is a vector containing deterministic components. $\mathbf{F_i} = (\mathbf{A'_{i1}}, \ldots, \mathbf{A'_{1p}}, \mathbf{\Phi'_i})'$ includes coefficient matrices. The error term $\boldsymbol{\varepsilon_t}$ is assumed to be independent normal with zero mean and variance-covariance matrix Ω. $\mathbb{G}_{\mathbf{t}}^{\mathbf{i}}(.)$ is a diagonal matrix of transition functions such that different transition functions across regimes can be modeled. Accordingly, $\mathbb{G}_{\mathbf{t}}^{\mathbf{i}}(.)$ reads:

$$\mathbb{G}_{\mathbf{t}}^{\mathbf{i}}(.) = \operatorname{diag}\big\{ \mathbf{g}(s_{1it}|\gamma_{i1}, c_{i1}), \ldots, \mathbf{g}(s_{kit}|\gamma_{ik}, c_{ik}) \big\} \tag{2}$$

for $i = 1 \ldots, m-1$, where m determines the number of transitions across equations and $\mathbb{G}_{\mathbf{t}}^{\mathbf{0}} = \mathbf{I_k}$, $\mathbb{G}_{\mathbf{t}}^{\mathbf{m}} = \mathbf{0}$. In the simulation exercise I stick to VSTAR models with a single transition ($m = 2$). The transition functions are assumed to be of logistic type which is monotonically increasing in s_{ijt}, where $j = 1, \ldots, k$, and bounded between zero and one:

$$\mathbf{g}(s_{ijt}|\gamma_{ij}, c_{ij}) = [1 + \exp(-\gamma_{ij}(s_{ijt} - c_{ij}))]^{-1}, \ \gamma_{ij} > 0 \tag{3}$$

The transition function depends on the transition speed (γ_{ij}), the location parameter (c_{ij}) and the transition variable (s_{ijt}). In order to make γ a scale-free parameter, it is divided by the standard deviation of the transition variable when the parameters of the VSTAR model are estimated following Teräsvirta (2004). The transition function which governs the transition from one regime to another is crucial in a Vector STAR model framework. In a multivariate framework, the type of transition function, transition variable, transition speed and the location parameter may be different in each equation which is often also economically reasonable. There is also the special case where only one transition function governs the whole system, then, $\mathbb{G}_{\mathbf{t}}^{\mathbf{i}}(.) = \mathbf{g}(s_{ijt}|\gamma_i, c_i)\mathbf{I_k}$. In the simulation exercises, it will be crucial to analyze different model specifications with respect to the transition function and parameter setting checking the performance of different starting-value search methods and their robustness across different specifications.

Amongst others van Dijk, Teräsvirta, and Franses (2002) discuss difficulties for estimating the slope parameter (γ_{ij}) of a Vector STAR model when the latter is large. When the transition speed is high, the Vector STAR model converges to a switching regression model. Determining the curvature might be problematic since a low number of observations around the location parameter could make the estimation of the slope parameter rather inaccurate in small samples. Thus, relying on suitable starting-values for the transition speed becomes even more important. The slope parameter γ_{ij} and thereby, the

[1] The notation is taken from Teräsvirta and Yang (2013).

Vector STAR model can be redefined by

$$\gamma_{ij} = \exp(\nu_{ij}), \tag{4}$$

where ν_{ij} is the parameter to be estimated. This methodological finesse has not been applied in the literature so far.[2] The slope parameter γ_{ij} in equation (3) can then be replaced by the expression in equation (4). To identify the model, $\gamma_{ij} > 0$ such that the codomain is restricted to be the set of positive real numbers. This implies that the redefinition is a bijective transformation. Redefining γ_{ij} facilitates the construction of the grid because one can build an equidistant grid in the dimension of ν_{ij}. Consequently, the search space (grid) for γ is automatically dense in the beginning and less so when it becomes large which is a sensible choice for estimating the Vector STAR model.

The estimation problem can be simplified by concentrating the likelihood function w.r.t. $\mathbf{F}$, as conditionally on $\mathbf{\Gamma} = [\gamma_{ij}]$, $\mathbf{C} = [c_{ij}]$ the model is linear (Leybourne, Newbold, and Vougas 1998). As a consequence, for the starting-value search the model can be either estimated by equation-by-equation OLS or system-wise by FGLS when it is efficient. The latter estimation approach is required if the multivariate model has zero restrictions in the lag structure and/or has cross-correlated error terms (Greene 2003: ch. 14).

Based on the derived initial values, the estimation is either carried out by an equation-by-equation Non-Linear Least Squares (NLS) algorithm or a system-wide Maximum-Likelihood (ML) approach. NLS minimizes equation-wise the residual sum of squares. The covariance matrix is estimated once at the end. Recall that initial values for the NLS estimation are obtained by OLS as described in the previous paragraph. In contrast to NLS, ML cannot be conducted equation-by-equation. Starting-values are obtained by the system estimation approach FGLS. ML does then also take the variance-covariance matrix ($\mathbf{\Omega}$) into account in the estimation. The numerical ML optimization w.r.t. $\boldsymbol{\gamma}$ and $\boldsymbol{c}$ takes $\mathbf{\Omega}$ as given. The covariance matrix is estimated in a previous step by FGLS. Whenever there are neither cross-correlations nor zero restrictions equation-by-equation NLS is efficient.

The estimation problem is bounded with respect to γ_{ij} and c_{ij}. The constraints are matched to the support of γ_{ij} and c_{ij} (see section 3 for more details).[3] For a more detailed description of specification, estimation and evaluation of Vector STAR models see Teräsvirta and Yang (2013) and for a survey on Vector Threshold and Vector Smooth Transition Autogressive Models see Hubrich and Teräsvirta (2013).

[2] I am indebted to Matt Holt and Timo Teräsvirta who suggested this approach.

[3] To optimize the Vector STAR model, I use an interior-point algorithm (ML estimation) and a trust-region-reflective algorithm (NLS estimation), both with constraints (lower and upper bounds) based on *fmincon.m* and *lsqnonlin.m* functions implemented in the MATLAB R2012b version.

3 Starting-Value Search Methods

In the following I present the competing starting-value search methods. In section 3.1, the grid search methods, the classical grid and the grid with a zoom, are presented. Subsequently in section 3.2, I introduce the heuristic methods which are threshold accepting, simulated annealing, and differential evolution.

3.1 Grid search

3.1.1 Classical grid

The classical grid search (GS) is based on the space of $\mathbf{\Gamma} = [\gamma_{ij}]$ and $\mathbf{C} = [c_{ij}]$ such that in higher dimensional models a multidimensional grid emerges assuming an equation-specific transition function. In order to make γ a scale-free parameter, following Teräsvirta (2004), it is divided by its standard deviation. The location parameter c_{ij} is a function of the transition speed γ_{ij}: $c_{ij}f(\gamma_{ij})$. If γ is high, implying a low number of observations around the threshold, we use a truncated sample of the observations of the transition variable for the location parameter c. At most, we exclude the lower and upper 15% percentile which is also recommended by Andrews (1993) and Caner and Hansen (2001) for Threshold (V)AR models. If γ is low, the support of c is not restricted. In other words, 100% of the transition variable observations are used as support. γ_{ij} is bounded between 0.1 and 30. It is necessary to constrain the parameter set as the VSTAR model becomes unidentified otherwise. The transition function is practically constant if either γ gets very large or small (or even negative). A slope parameter (γ) equal to 30 is already close to an abruptly switching Threshold VAR model. Recall that the slope parameter is redefined by $\gamma = \exp(\nu)$ for facilitating the starting-value search. This yields the new set $\mathbf{N} = [\nu_{ij}]$. Consequently, the bounds are redefined to $\mathbf{N} = \ln(\mathbf{\Gamma})$. For the new parameter set I use an equidistant search space with increments of 0.003, yielding an equidistant grid which is dense for low values and less so for steeper regions. The increments are chosen to match approximately the average number of likelihood evaluations of the heuristic starting-value search methods.

3.1.2 Grid with a zoom

The number of grid points increases quickly with model dimension and number of transition functions. Hence, it is likely that the number of grid points are intractable to estimate in a reasonable computing time if a multivariate approach is required. To circumvent the time-consuming grid search in higher dimensional models, Teräsvirta and Yang (2013) suggest a grid with a zoom which builds new grids by using the best solution of the previous step as the center. The initial step is a grid with a rather moderate

number of grid points in each dimension.[4] In the next step, the new grid is based on the neighboring points of the previously optimal solution. The zoom-in is discontinued when for all parameters ($\mathbf{\Gamma}$ and $\mathbf{C}$) the difference between the highest and smallest value building the new grid is smaller than a given value. 0.001 will be used in the following. The stepwise refinement and zoom-in have the ability to find a global optimum if the likelihood is centered around one global optimum and does not have many local optima. However, the grid with a zoom might eventually miss the global or a useful local optimum, if the surface area is not well-behaved. The zoom-in with a moderate grid clearly leads to a lower number of likelihood evaluation in contrast to heuristics. But since the heuristic methods are equipped with a stopping criterion which exits the algorithm earlier, they may have lower amount as well (see next section). In principle, the grid search and heuristics are equally efficient regarding their computational load. What makes the multivariate approach time-consuming is the FGLS estimation, where the covariance-matrix and its inverse are calculated, respectively. Hence, the computational load depends on the amount of function evaluations due to the covariance matrix calculation in order to derive the likelihood. To match the number of likelihood evaluations, one could also increase the number of grid points. Yet, we could then lose the time advantage of the grid with zoom. Moreover, increasing the number of grid points of the zoom-in may not necessarily result in superior outcomes because of the potential inability of this approach to find a global optimum of a non-smooth surface.

3.2 Heuristic Optimization Algorithms

In the class of optimization heuristics, I focus on local search methods which iteratively search for a new solution in the neighborhood of the current solution by using a random mechanism. These can be divided into two categories: population based methods and trajectory methods. The central idea of the local search methods is to allow temporary uphill/downhill moves, i.e. a (controlled) impairment of the value of the objective function.[5] This is done in order to escape local optima. These algorithms start off with a random guess such that they do not depend on individual subjective elements. Since we restart each algorithm, we in fact start with more than one random guess. In case of multiple local optima, heuristic methods may find better optima than traditional methods.
I assess the performance of different heuristic optimization algorithms in order to obtain a successful, fast and easily applicable modeling strategy. The starting-values search within a multivariate Vector STAR model relies on a continuous search space. Differential evolution (DE) which belongs to the class of population based methods might be preferable in this case (Gilli and Winker 2009). These kinds of method update a set of solutions si-

[4] I use five as recommended by Teräsvirta and Yang (2013).

[5] In a maximization problem, the methods do allow for downhill moves and vice versa in a minimization problem.

multaneously. Additionally, we employ threshold accepting (TA) and simulated annealing (SA). They are trajectory methods that work on a single solution, i.e. these procedures alter the value of only one parameter in each iteration step. These heuristic methods are described in more detail in the next sections beginning with trajectory methods.

3.2.1 Threshold Accepting

This section describes TA.[6] Algorithms 1 and 2 show pseudocodes for a minimization problem. First, the number of iterations I and restarts R, and the thresholds sequence (T) are initialized, see line 1 of Algorithm 1. The threshold sequence which is used for the acceptance decision gets linearly lowered to zero within 80% of the iterations. It is based on a data-driven threshold sequence which is endogenously generated from the sample (see Winker and Fang (1997) for details).[7] Line 2 shows the initialization of the (equation-specific) parameters of the transition speed and location parameter. The initial slope parameter (γ) of the transition function is drawn from an exponential distribution function with expected value one ensuring a positive value.[8] The initialized values of γ are transformed and the search procedure is based on $\nu = ln(\gamma)$. The initial location parameter (c) is drawn randomly from the actual realization of the (equation-specific) transition variable. As described above, c is a function of ν ensuring a feasible support of c based on the actual value of ν. When updating the parameters, the slope and location parameter are forced to remain in predefined bounds within the algorithm. When γ (ν) and c do not lie in the interval, they are set equal to their predefined bounds: 0.1 (ln(0.1)) and 30 (ln(30)) and lowest/highest value of the feasible range of the transition variable, respectively. Based on the initial parametrization, an equation-by-equation OLS estimation or system FGLS-estimation is performed. As can be seen in line 3 of Algorithm 1, an objective function – error variance or loglikelihood value – is calculated to compare different models. This value is stored.

After the initialization, the algorithm starts (line 4). The neighbor solution is computed in the next step, see line 5 in Algorithm 1. The following details can be found in Algorithm 2. A random draw determines whether the location or slope parameter is changed and in the multivariate set-up for which equation it is changed.[9] A random normal distributed term is added to the current value. In line with Maringer and Meyer (2008) the normal distribution has expected value of zero to allow for movements in both directions (positive and negative). The variance σ for ν is one, whereas the empirical variance of the transition

[6] The parameter setting of the algorithms is partly based on Maringer and Meyer (2008).

[7] An on-the-fly-updating, i.e. an updating during the iterations by taking local differences of the previously generated differences of the target functions of the threshold sequence as suggested by Lyra, Paha, Paterlini, and Winker (2010), leads to slightly inferior results on average. Hence, we stick to the data-driven threshold sequence.

[8] Preliminary experiments indicate that values of 2 or 3 do not lead to superior results.

[9] Preliminary experiments have shown that changing more than one parameter leads to inferior outcomes.

Algorithm 1 Pseudocode for Threshold Accepting Algorithm

1: Initialize (data-driven) threshold sequence T, number of iterations I and restarts R
2: Initialize $\Psi = (\mathbf{c}, \gamma)$: $\gamma = \exp(\mu = 1)$, $\mathbf{c}$ = uniformrand(range of transition variable), transform $\nu = ln(\gamma)$
3: Calculate current value of target function $f(\Psi)$
4: **for** $i = 1 : I$ **do**
5: Compute neighbor $\Psi^* \in \mathcal{N}(\Psi)$
6: Calculate $f(\Psi^*)$, $\Delta f = f(\Psi^*) - f(\Psi)$
7: **if** $\Delta f < T(i)$ **then**
8: keep modifications
9: **else**
10: undo modifications and keep previous solution
11: **end if**
12: Report elitist, lower threshold
13: **end for**

I=100,000 in the equation-by-equation approach (rounds=200, steps=500) and R=5;
I=500,000 in the multivariate approach (rounds=500, steps=1000) and R=3.

Algorithm 2 Neighbor for Threshold Accepting and Simulated Annealing Algorithm

1: Compute neighbor $\Psi^* \in \mathcal{N}(\Psi)$
2: **if** uniformrand$(0, 1) < 0.5$ **then**
3: Add $n1 = No(\mu_n = 0, \sigma_\nu = 1)$ to ν (for randomly selected equation)
4: **else**
5: Add $n2 = No(\mu_n = 0, \sigma_c = \text{std}(s_t))$ to c (for randomly selected equation)
6: **end if**

variable s_t is chosen for c. For the multivariate search procedure, σ_c is multiplied by two which leads to better results than using only the standard deviation itself.[10] Based on the updated parameter setting the Vector STAR model is estimated and the model fit – the error variance or loglikelihood – is calculated as displayed in line 6 of Algorithm 1. The difference of the objective function values between the previous and the new solution is calculated. The acceptance criterion is shown in line 7: if the difference is smaller than the current value of the threshold sequence, the new parameter setting is accepted else the previous solution is restored. In each iteration step, the best solution, that is called elitist, is kept. The next iteration steps follow until the predefined number of iterations have been carried out. They amount to 100,000 (rounds=500,steps=200) in the equation-by-equation and to 500,000 (rounds=1000,steps=500) for the system approach. The former (latter) is based on 5 (3) restarts of the algorithm. If within a predefined number of iterations ((rounds $\times$ steps)/10) the parameter combination and objective function value remain identical, implying that no improvement is achieved, the algorithm will be stopped. This implies assuming that a sufficiently good optimum is reached.

[10] In preliminary experiments, different values of the standard deviation or a downscaling of the standard deviation within the iterations have not resulted in superior outcomes. I assume that a normal distribution is more appropriate to overcome local optima than a uniform distribution. If we rely on a uniform distribution, we would get large changes with equally high probability as of small changes. This is not preferable. We favor small changes in most of the iteration steps and in few cases larger changes to overcome local minima. If we restricted the interval to a smaller area, we would exclude larger changes.

3.2.2 Simulated Annealing

Simulated annealing works in the same way as TA, except that the acceptance condition (line 7 of Algorithm 1) is replaced by the following expression:
$\Delta f < 0 \ \vee \ \exp(-\Delta f/\text{Temp}) > u$, where u is a uniformly (0,1) distributed pseudorandom variable. Hence, the acceptance rule becomes stochastic. Improvements of the objective function value are always accepted. The temperature (Temp), which is the relevant parameter for the acceptance condition, gets lowered during the iteration, what makes the acceptance of impairments less likely in the course of iterations. The parameter which governs the reduction of the temperature is called cooling parameter (cp). Instead of "lower threshold" in line 12, the expression reads: Temp = Temp $\times$ cp. The initial temperature is set equal to 10 and the cooling parameter is derived by following formula $\left(\frac{0.0001}{10}^{(1/\text{rounds})} = cp\right)$.[11] The formula ensures that the temperature is lowered until it is close to zero, that is here 0.0001.

3.2.3 Differential Evolution

In a multivariate non-linear Vector STAR model, it could be beneficial that the whole set of solutions (this is called a population) is updated simultaneously. Since the number of optimization parameters (γ and c) increases with model dimension and number of transitions, this approach may create an advantage in terms of velocity and efficiency. In contrast to trajectory methods, DE is more appropriate for a continuous search space. A pseudocode can be found in Algorithm 3.
There exist two main features which are important in the implementation of a differential evolution algorithm: mutation (slightly altering a solution) and cross-over (combination of the properties of two or more existing solutions). The former is determined by the scaling (weighting) factor F and the latter by the cross-over probability Π. As can be seen from lines 1 and 2 of Algorithm 3, both have to be initialized along with the population size n_p and the optimization parameters (γ and c). The latter two are initialized in line with the procedure of SA and TA. After the initialization, the value of the objective function is calculated for all potential solutions, see line 3. Then, the generations start off in line 4. The number of the generations (n_g) is chosen to match the number of objective function evaluations in SA and TA.
A candidate solution is constructed by taking the difference between two other solutions (members of the population), weighting this by a scalar F and adding it to a third solution as described in lines 7 to 9. Hence, F determines the speed of shrinkage in exploring the search space. Subsequently, an elementwise cross-over takes place across the intermediate and the original (existing) solution. This cross-over is determined by Π which is the probability of selecting either the original or the intermediate solution to

[11] Preliminary experiments have shown that a temperature of 10 is a good number to deal with the trade-off of escaping local optima and speed of convergence in this application.

Algorithm 3 Pseudocode for Differential Evolution

1: Initialize generations of population n_g, scaling factor $F = .8$ and cross-over probability $\Pi = .6$
2: Initialize population $pop = 10d$ by $\Psi = (\mathbf{c}, \gamma)$: $\gamma = \exp(\mu = 1)$, $\mathbf{c}$ = uniformrand(range of transition variable), transform $\nu = ln(\gamma)$
3: Calculate current value of target function $f(\Psi)$
4: **for** $i = 1 : n_g$ **do**
5: $P^0 = P^1$
6: **for** $i = 1 : n_p$ **do**
7: Select jth element of population p with dimension d (all parameters)
8: Generate interim solution by 3 distinct members (m1,m2,m3) of current $pop \neq j$
9: Compute interim solutions of parameters by $P^{int} = P^0_{m1} + F \times (P^0_{m1} + P^0_{m1})$
10: Generate offspring solution (P^1) by selecting with probability Π the parameter value from the interim or with $1 - \Pi$ from original population
11: Compute objective value
12: **if** $f(P^1) < f(P^0)$ **then**
13: Replace original solution by offspring solution, keep elitist
14: **end if**
15: **end for**
16: **end for**

n_g=100,000/(pop) and n_g=500,000/(pop) in system approach;
d=number of parameters being optimized.

form the offspring solution (see line 10).

To obtain the optimal combination of the population size n_p, the scaling parameter F and the cross-over probability Π, I performed pretests based on 100 repetitions. This three-dimensional grid shows a rather clear picture concerning the best parameter settings. The scaling parameter F should be at least 0.6. After F exceeds 0.5, the surface of the objective function shows a jump (steep downward trend) towards a lower (=better) value. After this threshold there are rather minor changes in the value of the objective function with a tendency of 0.8 being the optimal value. The cross-over probability Π does not yield such a clear picture concerning the optimality as the scaling parameter does, but the results indicate that it should be at least equal to 0.6. The algorithm chooses randomly from a finite set of solutions which it mixes what is called population. The population should be sufficiently large to allow for diversification such that a broad range of the search space is covered. Yet, it should not be too large to search efficiently through the search space finding the best solution. In this experiment, a population size greater than eight times the number of parameter delivers better results than smaller population sizes. I choose 10 as the multiplier for the number of parameters to get the population size. These results hold for all DGPs independent of zero restrictions or estimation approaches.

The acceptance condition in lines 12–14 is as follows: If the offspring solution results in a superior objective function, it replaces the existing solution. By construction the best solution (elitist) is always maintained in the population. If all parameters of the population are identical, the algorithm has converged and it will be stopped.

4 Assessment in a Monte Carlo experiment

4.1 Evaluation approach

The evaluation of the previously introduced starting-value search methods is based on a Monte Carlo experiment. I simulate DGPs of Vector STAR models in order to assess the performance of different search techniques by comparing the values of an objective function– the error variance or the loglikelihood.[12] The error variance is taken as target function if an equation-by-equation approach is efficient, whereas the loglikelihood is used for a multivariate search procedure.

The assessment of the starting-value search methods relies on two measures which are based on a pairwise comparison across procedures. The first measure calculates the mean and standard deviation of the absolute differences of target functions across procedures over all simulation runs ($\rightarrow$ **measure of similarity**). The second measure is the frequency of superior results over all simulation runs ($\rightarrow$ **measure of superiority**). An outcome is defined to be better than another if (i) the error variance is at least 0.05 per cent smaller than the error variance generated by the other algorithm or (ii) the loglikelihood is 0.05 per cent larger. Whenever the differences across procedures are considerable, I also assess which algorithm yields the best outcome across all procedures and which results in the best distribution of objective function values. The comparison across procedures is carried out for the results of both the starting-value search and the final estimation outcome. Restricting the evaluations to the results of the starting-value search is not sufficient. Two identical objective function values could refer to different optima. After the optimization, the final value of the objective function could then be different.[13]

4.2 Data Generating Processes of VSTAR models

The Monte Carlo simulations rely on 5000 replications for the equation-by-equation approach and 1000 replications for the multivariate search procedure. The sample size is $T = 250$, which corresponds to approximately 20 years of monthly data. This defines a finite sample setting. Time series with length $T + 100$ are generated, and the initial 100 observations are discarded to eliminate the dependence on the initial value (seed). The error terms $\varepsilon_{\mathbf{t}}$ are drawn from a normal distribution with expected value zero and variance-covariance matrix (Ω) [$\varepsilon_{\mathbf{t}} \sim N(0, \Omega)$].

The Vector STAR models to be simulated can be found in Table 1. The first Vector STAR model (VSTAR1) relies on a lag structure without "gaps", whereas the second (VSTAR2)

[12] I take the lag structure as given in the starting-value search and the optimization. Hence, the optimization problem relies on a "known" lag structure.

[13] I do not compare the likelihood to the "true" likelihood of the DGP. First, the DGPs are based on a finite sample which does neither allow to compare exact parameter estimates nor might necessarily yield the what is commonly called "true" likelihood. Second, I seek to find the "best" implementation for a given estimator for which no analytical solution exists.

contains zero restrictions. The degree of non-linearity and the (non-)stationarity of the process is determined by the parameters which are chosen such that the process seems to be stable and does not exhibit explosive behavior.[14] For a non-linear, multivariate Vector STAR model the stationary conditions have not been theoretically derived so far.
In the simulation exercise the equation-specific location parameter (c) is set to values which are close to zero. This reflects a reasonable magnitude with respect to economic applications assuming regimes which are related to boom and bust scenarios, for instance positive and negative output growth. By setting $\gamma = [3, 2]$ the transition speed is chosen such that a moderate transition speed emerges rather than a linear model or an abrupt change (VSTAR1-1 and VSTAR2-1). Yet, I also model a case in which a rather abrupt change takes place, where $\gamma = 20$ for the first equation (VSTAR1-2 and VSTAR2-2). The probability that a value of the transition function lies in the open interval between 0.01 and 0.99 is a measure for the steepness of the function, hence, the transition speed. The larger the probability, the smoother the function. For $\gamma = [3, 2]$ it amounts to 80.6% and 95.1% on average. Choosing $\gamma = 20$, the values of the transition function are more frequently closer to 0 or 1 as the probability of lying between 0.01 and 0.99 is only approximately 15%.
VSTAR1-3 and VSTAR2-3 have cross-correlated error terms, otherwise being identical to VSTAR1-1 and VSTAR2-1. Finally, the third (VSTAR3) model is a trivariate VSTAR process. VSTAR3-1 has equation-specific transition functions, whereas a single transition function governs the VSTAR3-2 model. For the former model the transition speed varies across equations, the probability of lying between 0.01 and 0.99 amounts to 99.6%, 91.5%, and 40.4% for $\gamma = [.5, 2, 7]$. For VSTAR3-2 the probability is 63.8% for $\gamma = 4$.
Theoretically, VSTAR1-1 and VSTAR1-2 can be efficiently estimated equation-by-equation by OLS (starting-value search) and NLS (optimization). All other VSTAR-DGPs require a multivariate search (FGLS) and optimization procedure (ML) due to a lag structure with zero restrictions, cross-correlated error terms and/or a single transition function governing the whole system. Nevertheless the empirical results might be different. Hence, I employ the equation-by-equation as well as the multivariate search procedures and estimation strategy for all DGPs to find the best implementation for the starting-value search.

5 Simulation Results

I begin by presenting the results of different starting-value search methods of the equation-by-equation approach in subsection 5.1. As mentioned before, in a Vector STAR model

[14] See equations (5), (6) and (7) in the appendix for the exact parameter specification of all VSTAR models. The Vector STAR VSTAR1-1, VSTAR1-2, VSTAR1-3, VSTAR3-1, and VSTAR3-2 processes are depicted in figures 3 – 7 in the appendix. There is visually no difference between VSTAR1 and VSTAR2, hence, I do not show the figures of VSTAR2.

Table 1: Simulated Vector STAR DGPs

	transition function	γ	c	d	Ω
VSTAR1-1	logistic	[3,2]	[0,0.5]	$\{1,2\}$	[1,0;0,1]
VSTAR1-2	logistic	[20,2]	[0,0.5]	$\{1,2\}$	[1,0;0,1]
VSTAR1-3	logistic	[3,2]	[0,0.5]	$\{1,2\}$	[1,1.5;1.5,3]
VSTAR2-1	logistic	[3,2]	[0,0.5]	$\{1,2\}$	[1,0;0,1]
VSTAR2-2	logistic	[20,2]	[0,0.5]	$\{1,2\}$	[1,0;0,1]
VSTAR2-3	logistic	[3,2]	[0,0.5]	$\{1,2\}$	[1,1.5;1.5,3]
VSTAR3-1	logistic	[.5,2,7]	[.5,0,.11]	$\{2,1,4\}$	[1,0,0;0,1,0;0,0,1]
VSTAR3-2	logistic	[4]	[0]	$\{1\}$	[1,0,0;0,1,0;0,0,1]

VSTAR1 and VSTAR2 differ w.r.t. their lag structure. VSTAR1 relies on a lag structure without "gaps", VSTAR2 contains zero restrictions.
VSTAR3 contains zero restrictions. All specification can be also found in the appendix.

without zero restrictions and no cross-correlations an equation-by-equation NLS estimation is efficient. Thus, the starting-value search can be based on equation-by-equation OLS.[15] The outcomes are assessed by conducting comparisons across procedures by means of the measures of similarity and superiority for both the starting-value search and the estimation outcomes as described in section 4.1. In subsection 5.2, I assess Vector STAR DPGs for which a system (multivariate) approach is efficient. I employ a multivariate starting-value search setting. Besides comparing the different procedures, I check whether in this setting the equation-by-equation approach indeed yields worse results.
For the sake of convenience, I will present results by using only the loglikelihood values. The equation-by-equation approach still takes the error variance as target function. Based on these results, I calculate the loglikelihood.

5.1 Equation-by-equation starting-value search

The results in Table 2 on the measure of similarity already indicates that all algorithms generate quite similar loglikelihood values of the starting-value search for VSTAR1-1 and VSTAR1-2. In particular, SA, TA and DE seem to perform equally well. This is also confirmed by the results of the measure of superiority in Table 3. GS delivers slightly worse results than the other methods. In approximately 4% of the simulation runs the heuristic algorithms generate a 0.05 per cent higher loglikelihood value than GS. On average, GS has a lower number of objective function evaluation than SA and TA does: 403,047 vs. 440,061 and 500,000. This could explain the slightly worse outcomes. Yet, the function evaluations of DE amount to 402,479 on average which is the lowest value

[15]This model set up differs from a univariate approach since the right hand side includes also lagged variables of the dependent variable of the second equation. The results do nonetheless hold for a univariate model set-up.

across all procedures. I therefore assume that the fixed grid points for the parameter values are responsible for the inferiority of the grid search. The heuristic methods allow the parameters in principle to take any value which then could easily result in a higher loglikelihood.

Table 2: VSTAR1-1 and VSTAR1-2, starting-value search – absolute differences

	DE/GS	DE/TA	DE/SA	SA/TA	SA/GS	TA/GS
	VSTAR1-1					
mean	0.0122	0.0010	0.0011	0.0001	0.0117	0.0118
std	0.0302	0.0176	0.0176	0.0006	0.0262	0.0262
	VSTAR1-2					
mean	0.0113	0.0007	0.0008	0.0001	0.0108	0.0108
std	0.0261	0.0140	0.0140	0.0008	0.0226	0.0227

Remark for interpretation: The absolute differences of the pairs yield a measure of similarity. The smaller the absolute difference, the more similar the loglikelihood of the algorithms.

Table 3: VSTAR1-1 and VSTAR1-2, starting-value search – frequency of superior results

	VSTAR1-1				VSTAR1-2			
	DE	SA	TA	GS	DE	SA	TA	GS
DE	-	0.10%	0.10%	4.42%	-	0.10%	0.10%	3.84%
SA	0.30%	-	0.00%	4.42%	0.24%	-	0.00%	3.74%
TA	0.30%	0.00%	-	4.44%	0.24%	0.00%	-	3.78%
GS	0.36%	0.18%	0.18%	-	0.32%	0.14%	0.14%	-

Remark for interpretation: Row better than column.
At least 0.05 per cent larger likelihood than other algorithm.

It is not sufficient to focus solely on the starting-value search outcomes. It is necessary to compare values of the objective function after the optimization as well. As mentioned before, in theory two differently located optima with the same objective function value could be found by the starting-value search. These could lead to different optimized values. First of all, I check whether an equation-by-equation NLS estimation or a system ML estimation generates better results. Depending on the outcome, I use either the results obtained by NLS or ML for the final assessment. The results in Table 4 show that the frequency of superior loglikelihood values is on average higher for NLS estimation. This is in line with NLS being efficient for this VSTAR model set-up as the models do neither exhibit zero restrictions nor cross-correlated error terms. The difference between both estimation procedures does not seem to be large, however. In 70% of the simulation runs the outcomes yield a loglikelihood value which does not differ by more than 0.05 per cent. To evaluate the results, I use the values obtained by NLS. Yet, the results discussed in the following hold for ML estimation as well.

After estimating the VSTAR1-1 and VSTAR1-2 model using the initial values obtained by the respective algorithms, the absolute differences of the target function value across

Table 4: VSTAR1-1 and VSTAR1-2, NLS vs. ML estimation – frequency of superior results

	NLS better than ML		ML better than NLS	
	VSTAR1-1	VSTAR1-2	VSTAR1-1	VSTAR1-2
DE	25.10%	23.78%	6.26%	6.38%
SA	24.96%	23.72%	6.38%	6.34%
TA	24.74%	23.58%	6.42%	6.38%
GS	27.74%	25.98%	5.16%	5.38%

Remark for interpretation: Better means at least 0.05 per cent larger likelihood.

procedures decrease further or remain approximately the same as can be seen from Tables 5 and 6. The value of the objective function is in almost all runs identical suggesting the detection of the same optimum and an identical performance in the equation-by-equation starting-value search setting. Small differences in starting-values do not have a large impact on final estimates. In particular, GS now yields as good results as the heuristic methods. Hence, the disadvantage of the rigid grid is offset after the estimation. The grid search comes already very close to the optimum obtained by the heuristics but an optimization algorithm is necessary to reach it.

Table 5: VSTAR1-1 and VSTAR1-2, NLS Estimation – absolute differences

	DE/GS	DE/TA	DE/SA	SA/TA	SA/GS	TA/GS
	VSTAR1-1					
mean	0.0045	0.0018	0.0021	0.0009	0.0044	0.0046
std	0.0648	0.0375	0.0445	0.0283	0.0681	0.0723
	VSTAR1-2					
mean	0.0050	0.0037	0.0037	0.0006	0.0024	0.0028
std	0.1049	0.1055	0.1051	0.0255	0.0392	0.0492

Remark for interpretation: The absolute differences of the pairs yield a measure of similarity. The smaller the absolute difference, the more similar the loglikelihood of the algorithms.

Table 6: VSTAR1-1 and VSTAR1-2, NLS Estimation – frequency of superior results

	VSTAR1-1				VSTAR1-2			
	DE	SA	TA	GS	DE	SA	TA	GS
DE	-	0.16%	0.26%	0.32%	-	0.22%	0.22%	0.40%
SA	0.22%	-	0.12%	0.30%	0.28%	-	0.04%	0.28%
TA	0.18%	0.02%	-	0.28%	0.28%	0.04%	-	0.28%
GS	0.48%	0.40%	0.48%	-	0.54%	0.32%	0.36%	-

Remark for interpretation: Row better than column.
Better means at least 0.05 per cent larger likelihood.

This is confirmed by Table 7 which shows the comparison of the starting-value search and optimization outcomes by counting the number (i) for which the optimized loglikelihood is at least 0.01 per cent better than that of the starting-value search and (ii) of

identical results of starting-value search and estimation (up to the fourth decimal). The results indicate that the starting-value search by heuristic algorithms frequently, in about 50–60% of all runs, produces the final optimum already. The number of improvements after estimating the Vector STAR model is clearly lower than those of GS. Nevertheless, improvements are possible, while DE seems to be a bit more efficient in the starting-value search. Overall, the grid search yields a worse approximation to the optimum and a gradient-based algorithm is necessary in order to obtain a better, hopefully global, optimum.

Table 7: VSTAR1-1 and VSTAR1-2, comparison of starting-value search and optimization

	(i)		(ii)	
	VSTAR1-1	VSTAR1-2	VSTAR1-1	VSTAR1-2
DE	38.14%	35.72%	58.42%	60.94%
SA	37.88%	35.48%	54.08%	56.84%
TA	37.78%	35.42%	54.26%	56.06%
GS	58.90%	55.98%	1.18%	0.94%

(i) Optimized loglikelihood at least 0.01 per cent better than starting-value search.
(ii) Identical results of estimation and starting-value search (up to the fourth decimal).

These results hold for Vector STAR DGPs without zero restrictions and cross-correlations across error terms within the equation-by-equation approach. To sum up, all starting-value search procedures work equally well in the equation-by-equation approach. The heuristics, DE somewhat more pronounced than the other, are already quite efficient in the starting-value search as long as the equation-by-equation approach is efficient. As can be seen from Figure 1, the parameter estimates for Γ and C confirm these outcomes as the curves are practically identical. The figure refers to the first equation for VSTAR1-1 but the outcomes for the other equation as well as VSTAR1-2 are similar. In more than 95% of the simulated cases, the parameter estimates do not differ by more than 0.01 across all procedures in the simulation runs. This is true for the starting-value search as well as for the optimization.

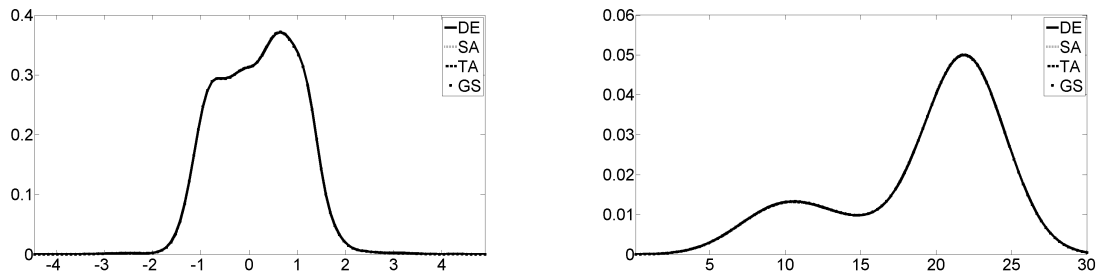

Figure 1: Kernel density of parameter estimates after starting-value search, normal kernel, bandwidth optimized for normal kernel – c and γ, VSTAR1-1 (true $c = 0.5$ and $\gamma = 2$, $T = 250$)

I also compare equation-by-equation search approaches and the multivariate search strategies although the former is efficient. As we are in a final sample set-up, a multivariate approach could nevertheless be beneficial. From Table 8 it is seen that the systemwise starting-value search is not significantly superior to the equation-by-equation approach. Rather the equation-by-equation search yields on average better results. Knowing that the latter is also much faster than the former, there is no need to use the multivariate starting-value search.[16] The equation-by-equation implementations of TA and GS seem to be particularly more efficient than its multivariate counterpart. In about 90% (GS) to almost 100% (TA) the equation-by-equation is better than the multivariate approach. In the following section 5.2, we will see that it is rather the ineffectiveness of the multivariate TA and GS algorithm which drives this result. For DE and SA the superiority of the equation-by-equation approach is lower (17% and 28%), but still larger than the percentage rate of the superiority of the multivariate over the equation-by-equation approach.

Table 8: VSTAR1-1 and VSTAR1-2, equation-by-equation vs. system approach

	VSTAR1-1				VSTAR1-2			
	Starting-value search							
	DE	SA	TA	GS	DE	SA	TA	GS
Eq.-eq.$\succ$Multiv.	17.40%	27.30%	99.70%	92.00%	18.60%	28.20%	99.90%	91.30%
Multiv.$\succ$Eq.-eq.	10.50%	9.10%	0.10%	1.90%	9.30%	9.20%	0.10%	1.70%
	NLS estimation							
	DE	SA	TA	GS	DE	SA	TA	GS
Eq.-eq.$\succ$Multiv.	37.80%	46.00%	58.30%	62.20%	38.00%	45.90%	56.70%	59.80%
Multiv.$\succ$Eq.-eq.	9.70%	8.30%	14.00%	11.30%	8.90%	8.30%	14.20%	12.00%

Remark for interpretation: Better ($\succ$) means at least 0.05 per cent larger likelihood.

When an equation-by-equation approach is not efficient or not possible at all due to a single transition function governing all equations, we have to estimate the model system-wise. Then, it becomes already infeasible to estimate a four dimensional grid in a reasonable computing time. Therefore, we rely on the grid with a zoom advocated by Teräsvirta and Yang (2013) as a benchmark for the heuristic algorithms to derive starting-values for a multivariate search and estimation strategy in the following section.

5.2 Multivariate starting-value search

There are three types of VSTAR models which theoretically require a multivariate search and estimation procedure: i) VSTAR2 and VSTAR3 with zero restrictions in the lag structure, ii) VSTAR1-3 and VSTAR2-3 with cross-correlated error terms, and iii) VSTAR3-2 which has a single transition function governing the whole model. In this section, I assess, the outcomes of different algorithm for finding starting-values in a multivariate setting.

[16]Recall that the estimation is done by NLS. All results presented above hold for the ML estimation procedures as well. Results are available upon request.

Besides that, I evaluate whether the multivariate search and estimation procedure indeed outperforms the equation-by-equation approach. This might not necessarily be the case in applications with short or moderately long time series. The results for VSTAR2-1 and VSTAR2-2 will be discussed in section 5.2.1 and for the trivariate VSTAR3-1 process in section 5.2.2. Section 5.2.3 shows the outcomes for VSTAR models with cross-correlated errors (VSTAR1-3 and VSTAR2-3), and section 5.2.4 those for VSTAR3-2 with a single transition function.

5.2.1 Bivariate VSTAR model with zero restrictions

I begin by discussing the outcomes of the algorithms applied to VSTAR2 model without cross-correlated errors. I employ both a multivariate and an equation-by-equation approach for the starting-value search. First of all, I check whether the equation-by-equation search should be followed by NLS or ML estimation. When it comes to the estimation, it is seen from Table 9 that the equation-by-equation approach and ML estimation on average yields slightly worse results than NLS estimation. Yet, in approximately 70% of the simulations the estimation procedures obtain a loglikelihood value which differs not more than 0.05 per cent. Based on those results, I take the equation-by-equation search with NLS estimation as a benchmark for the comparison with the multivariate approach to be on the safe side. If the multivariate approach outperforms the former followed by NLS estimation, the probability of outperforming ML will be even higher.

Table 9: VSTAR2-1 and VSTAR2-2, NLS vs. ML estimation – frequency of superior results

	NLS better than ML		ML better than NLS	
	VSTAR2-1	VSTAR2-2	VSTAR2-1	VSTAR2-2
DE	22.44%	20.90%	9.26%	8.70%
SA	22.20%	20.68%	9.44%	8.80%
TA	22.16%	20.56%	9.42%	8.98%
GS	23.82%	22.58%	7.96%	7.54%

Remark for interpretation: Better means at least 0.05 per cent larger likelihood.

The results in Table 10 show that although the multivariate approach would be efficient, the multivariate search with ML estimation does not yield clearly better outcomes on average than the equation-by-equation search associated with NLS. This is particularly true for TA and GS, whereas for SA and DE the superiority is less pronounced. Yet, for the latter two procedures the equation-by-equation approach yields even better results after Vector STAR model estimation. Looking at the difference between multivariate and equation-by-equation search in more detail, the difference increases for DE from roughly 3%–7% to 16–23% and for SA from approximately 25% to 30%. Hence, there is already a tendency for the equation-by-equation starting-value approach to perform better than the system-wise search. This becomes even more obvious after estimating the

Vector STAR model. The equation-by-equation approach generates a higher frequency of superior results for DE and SA than the system search. The clear superiority of the equation-by-equation approach does hold for the starting-value search, but decreases for Vector STAR model estimation for TA and GS. This is once more an indication that the multivariate approaches of TA and GS does not seem to be very efficient. From these results, there can be drawn a rather clear conclusion which optimization strategy to apply. The gain of efficiency of estimating VSTAR models with zero restrictions system-wise is not pronounced in this application. Even if the results were completely identical, the equation-by-equation approach would be preferable due to a shorter execution time.[17] Eventually, one could force the multivariate approach to generate better results if one increased the number of likelihood evaluations. This, however, is not in line with the aim of the study which is to provide an easily applicable modeling strategy. The equation-by-equation methods seem to search more effectively through the parameter space than their system-wise counterparts. The error of not taking the covariance matrix into account in the search procedure is negligible. From the applied point of view, this is a useful result.

Table 10: VSTAR2-1 and VSTAR2-2, equation-by-equation vs. system approach

	VSTAR2-1				VSTAR2-2			
	DE	SA	TA	GS	DE	SA	TA	GS
	Starting-value search							
Eq.-eq.$\succ$Multiv.	15.80%	23.50%	99.10%	89.50%	18.10%	25.10%	99.70%	88.50%
Multiv.$\succ$Eq.-eq.	12.50%	10.60%	0.00%	1.20%	11.00%	9.40%	0.00%	1.90%
	Estimation							
Eq.-eq.$\succ$Multiv.	33.30%	39.70%	57.20%	59.40%	21.20%	29.30%	54.40%	55.10%
Multiv.$\succ$Eq.-eq.	10.80%	9.60%	14.30%	13.00%	5.10%	4.20%	18.60%	16.60%

Remark for interpretation: Better ($\succ$) means at least 0.05 per cent larger likelihood.

Next, we assess the different equation-by-equation starting-value search methods by pairwise comparisons. To begin with, the results in Table 11 support those shown in Section 5.1 for the equation-by-equation procedure applied to VSTAR1. The algorithms for finding starting-values do not yield strong differences, where SA, TA and DE slightly outperform GS in the starting-value search. After the optimization, the outcomes also show the same pattern as in the previous section. In principle, all methods could be used to find good initial values for bivariate Vector STAR models that contain zero restrictions but no cross-correlated error terms.[18]

In order to analyze the effectiveness of the starting-value search procedure, I compare the values obtained by starting values search and after estimation in Table 12. The heuristics yield a high percentage rate of identical results after starting-value search and estimation

[17] Recall that we rely on equation-by-equation OLS estimation which is much faster than the multivariate procedure associated with FGLS estimation.

[18] Identically to VSTAR1-1 and -2, the parameter estimates of Γ and C across procedures show marginal differences in VSTAR2-1 and -2.

Table 11: VSTAR2-1 and VSTAR2-2, equation-by-equation search – frequency of superior loglikelihood

	VSTAR2-1				VSTAR2-2			
	Starting-value search							
	DE	SA	TA	GS	DE	SA	TA	GS
DE	-	0.12%	0.12%	2.64%	-	0.12%	0.12%	3.48%
SA	0.30%	-	0.00%	2.68%	0.34%	-	0.02%	3.36%
TA	0.30%	0.00%	-	2.68%	0.36%	0.02%	-	3.42%
GS	0.38%	0.22%	0.22%	-	0.42%	0.12%	0.14%	-
	NLS estimation							
DE	-	0.20%	0.46%	0.20%	-	0.20%	0.36%	0.36%
SA	0.28%	-	0.26%	0.16%	0.20%	-	0.20%	0.26%
TA	0.28%	0.00%	-	0.16%	0.18%	0.02%	-	0.24%
GS	0.40%	0.34%	0.58%	-	0.40%	0.32%	0.44%	-

Remark for interpretation: Row better than column.
At least 0.05 per cent larger likelihood than other algorithm.

(up to the fourth decimal). This shows the efficiency of the heuristics already in the starting-value search. In case of the grid search, the optimized loglikelihood is in more than 50% of the simulation runs larger than that generated by the initial search. In less than 2% the results of estimation and starting-value search are identical up to the fourth decimal.

Table 12: VSTAR2-1 and VSTAR2-2, comparison of starting-value search and optimization

	(i)		(ii)	
	VSTAR2-1	VSTAR2-2	VSTAR2-1	VSTAR2-2
DE	34.02%	29.92%	62.44%	67.30%
SA	33.64%	29.44%	58.48%	63.62%
TA	33.24%	29.22%	58.22%	62.90%
GS	51.56%	49.40%	1.86%	1.64%

(i) Optimized loglikelihood at least 0.01 per cent better than starting-value search.
(ii) Identical results of estimation and starting-value search (up to the fourth decimal).

Hence, we can conclude that the gain of the derivative-based algorithm is larger. Yet, after the derivative-based optimization algorithm is carried out, the difference across the values of different procedures is reduced, implying that GS is able to obtain the same optimum as the heuristics.

5.2.2 Trivariate VSTAR model with zero restrictions

Before I present the results of the VSTAR models with cross-correlations in section 5.2.3, I consider a higher-dimensional VSTAR model with zero restrictions. Particulary, I focus on a trivariate VSTAR model (VSTAR3-1) with equation-specific transition functions.

The results in Table 13 for VSTAR3-1 differ to some extent from the ones shown before. First of all, ML estimation is on average clearly better than NLS estimation after the equation-by-equation search. Consequently, a more complex process (higher dimension) with zero restrictions requires and benefits from a system-wide derivative-based algorithm which takes the covariance matrix into account (ML estimation). This has not necessarily been the case for the bivariate processes.

Table 13: VSTAR3-1, NLS vs. ML estimation – frequency of superior results

	ML better than NLS	NLS better than ML
DE	56.90%	19.24%
SA	57.10%	18.96%
TA	57.12%	18.90%
GS	54.54%	19.70%

Remark for interpretation: At least 0.05 per cent larger likelihood than other algorithm.

Given these results, I compare the equation-by-equation and the multivariate search procedures, both associated with ML estimation. From Table 14 it is seen that regarding the starting-value search, the equation-by-equation approach is preferable for all algorithms. The advantage becomes smaller after the ML estimation, but still for TA and SA a clear superiority of the equation-by-equation approach is maintained. For DE and GS the multivariate search is only slightly better on average. The distinction is comparatively small which might not be seen as a clear indication for the equation-by-equation search. Yet, having in mind that the multivariate procedure is associated with a higher computational load, equation-by-equation search appears preferable.

Table 14: VSTAR3-1 – equation-by-equation vs. system approach

	Starting-value search			
	DE	SA	TA	GS
Eq.-eq.$\succ$Multiv.	74.90%	87.90%	100.00%	89.30%
Multiv.$\succ$Eq.-eq.	11.60%	6.80%	0.00%	7.00%
	ML estimation			
Eq.-eq.$\succ$Multiv.	42.40%	48.70%	37.40%	32.70%
Multiv.$\succ$Eq.-eq.	37.60%	31.20%	21.60%	26.90%

Remark for interpretation: Better ($\succ$) means at least 0.05 per cent larger likelihood.

This is again a useful result from an applied perspective: even a more complex model does not necessarily require a multivariate, time-consuming search procedure. Equation-by-equation starting-value search obtains better or sufficiently good starting-values. It is even evident that equation-by-equation search is superior.

Table 15 contains results for the individual algorithms. They indicate that the heuristics perform slightly better in the starting-value search than the grid search. But then, in

contrast to the results derived in the previous section 5.2.1, this advantage does not become negligible after ML estimation. The grid search does not find the best optimum after ML estimation although the inferiority is about 4–5% which is not extremely high. Obviously, the inflexibility of the grid in contrast to the heuristic search space is not compensated by an optimization algorithm in a more complex, trivariate process as it has been for a bivariate model.

Table 15: VSTAR3-1, equation-by-equation search – frequency of superior loglikelihood

	Starting-value search				ML estimation			
	DE	SA	TA	GS	DE	SA	TA	GS
DE	-	0.14%	0.14%	5.74%	-	0.30%	0.30%	4.14%
SA	0.50%	-	0.00%	5.72%	1.12%	-	0.40%	4.62%
TA	0.50%	0.00%	-	5.72%	1.94%	1.20%	-	5.38%
GS	0.72%	0.30%	0.30%	-	1.72%	1.40%	1.32%	-

Remark for interpretation: Row better than column.
At least 0.05 per cent larger likelihood than other algorithm.

To sum up, even when the dimension of the model is increased from two to three, there is no clear winner of the starting-value search procedure across heuristics. Yet, the GS performs slightly worse than the latter. The results differ from the previous ones in section 5.2.1 in three respects. First, the less complex processes yield on average better outcomes using NLS estimation than ML, whereas ML estimation results in higher loglikelihood values than NLS for the trivariate model. Second, GS yields slightly worse results also for the final estimates. Third, a derivative-based algorithm is clearly necessary to improve the loglikelihood derived by the starting-value search which can be seen from Table 16. In around 73% of the simulation runs the optimized value is better than the loglikelihood value obtained by the starting-value search. This is independent from the procedure used. This indicates that although the search problem becomes more complex, an equation-by-equation search equipped with a derivative-based algorithm can handle this optimization problem sufficiently well.

Table 16: VSTAR3-1, Comparison of starting-value search and optimization

	(i)	ii)
DE	72.94%	0.00%
SA	73.02%	0.02%
TA	73.02%	0.00%
GS	73.42%	0.02%

(i) Optimized loglikelihood at least 0.01 per cent better than starting-value search.
(ii) Identical results of estimation and starting-value search (up to the fourth decimal).

5.2.3 Bivariate VSTAR model with cross-correlated errors

The second type of VSTAR models which formally require a multivariate search and estimation procedure are those with cross-correlated error terms (VSTAR1-3 and VSTAR2-3). As can be seen directly from Table 17, the equation-by-equation search procedure is not superior anymore in this setting. This is in contrast to the processes that only contain zero restrictions. The system approach outperforms the equation-by-equation search in almost every simulation run. Consequently, the effectiveness of a method for a VSTAR model with cross-correlated errors is clearly reduced if the covariance-matrix is not taken into account in the starting-value search.

Table 17: VSTAR3-1 and VSTAR3-2, equation-by-equation vs. system approach

	VSTAR1-3				VSTAR2-3			
	Starting-value search							
	DE	SA	TA	GS	DE	SA	TA	GS
Eq.-eq.$\succ$Multiv.	0.00%	0.00%	1.00%	0.50%	0.00%	0.00%	3.50%	0.00%
Multiv.$\succ$Eq.-eq.	100.00%	100.00%	99.00%	99.50%	100.00%	100.00%	96.50%	100.00%
	ML estimation							
Eq.-eq.$\succ$Multiv.	1.60%	2.60%	12.60%	11.00%	0.80%	2.00%	11.50%	3.70%
Multiv.$\succ$Eq.-eq.	95.20%	94.20%	82.30%	84.50%	94.00%	92.80%	81.40%	89.50%

Remark for interpretation: Better ($\succ$) means at least 0.05 per cent larger likelihood.

After the ML estimation, the superiority of the system-wide search is still significantly evident for DE and SA, although the magnitude decreases. The margin of GS and TA between the equation-by-equation and the multivariate search gets more narrow, but is still of considerable magnitude. Turning this around, however, the equation-by-equation performance is clearly worse than the system-wide approach for all methods. The advantage of GS with a zoom over the standard GS also decreases somewhat more after the estimation than that of SA and DE. On the one hand, the number of likelihood evaluations for a grid with a zoom are clearly lower than of the heuristics, but on the other hand the zoom-in may lead to an inferior local optimum. This argument will be discussed later on.

When it comes to the performance of the different methods, the multivariate starting-value search yields different results than the equation-by-equation approach across procedures. This is obvious from Table 18. The measure of similarity clearly increases showing diverging results between the search procedures.

This is also seen from Figure 2 by considering the parameter estimates (γ and c) that vary from one algorithm to the next with the exception of SA and DE. The latter procedures yield quite similar parameter estimates. This is in contrast to the results from the previous section where besides the loglikelihood values also the final parameter estimates coincide. Table 19 reports values of the measure of superiority. They suggest that TA is inferior

Table 18: VSTAR3-1 and VSTAR3-2, absolute differences of loglikelihood

	starting-value search					
	DE/GS	DE/TA	DE/SA	SA/TA	SA/GS	TA/GS
	VSTAR1-3					
mean	1.6488	3.3969	0.0616	3.3629	1.6173	2.1763
std	1.7006	1.8777	0.0924	1.8490	1.6832	1.7406
	VSTAR2-3					
mean	0.96965	5.32360	0.09963	5.24100	0.92716	4.47549
std	1.32375	3.26375	0.12747	3.22714	1.29228	3.33831
	ML Estimation					
	VSTAR1-3					
mean	0.8341	1.0595	0.0656	1.0588	0.8322	1.0735
std	1.3981	1.5094	0.1413	1.4964	1.3837	1.6529
	VSTAR2-3					
mean	0.38956	1.06746	0.07501	1.07126	0.39921	1.04292
std	0.89976	2.13514	0.11615	2.09957	0.88911	2.16230

Remark for interpretation: The absolute differences of the pairs yield a measure of similarity. The smaller the absolute difference, the more similar the loglikelihood of the algorithms.

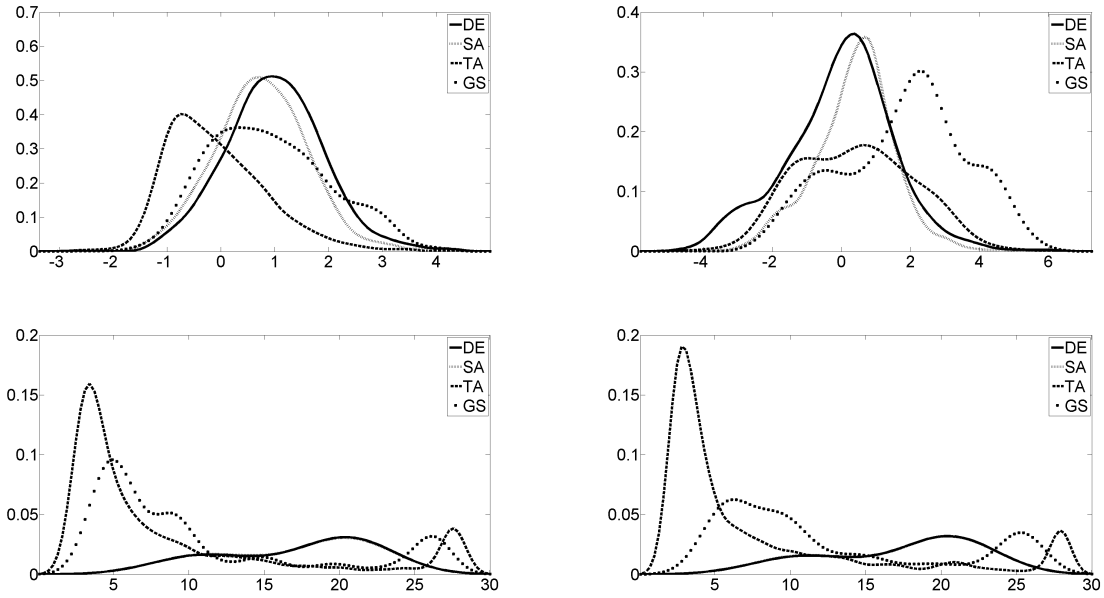

Figure 2: Kernel density of parameter estimates after starting-value search, normal kernel, bandwidth optimized for normal kernel – c (upper graphs) and γ (lower graphs), VSTAR1-3 (true $c = [0, 0.5]$ and $\gamma = [3, 2]$, $T = 250$)

to the other algorithms. In almost all simulation runs it generates lower loglikelihood values than the other algorithms. Although the grid with a zoom does not yield very convincing results either, it still beats TA in 80%–95%. However, the grid with a zoom is also clearly outperformed by DE and SA in 70%–90% of the simulations runs. DE in particular seems to be very efficient. It can be regarded as the best starting-value search procedure assessing solely the loglikelihood values of the starting-value search procedures. This result still holds after the optimization which is the meaningful statistic. Yet, the clearness of the results somewhat decreases. TA remains the worst starting-value search procedure and does not converge to a good optimum. The search procedure does not seem to be very effective which may be due to an inefficient acceptance criterion. This could come from a threshold sequence which may not allow for large impairments. Hence, the final outcome could be then a local optimum. DE and SA still perform better than GS grid with a zoom in around 32%–50% of the simulation runs, whereas GS features a clearly lower frequency of superior results w.r.t. SA and DE (only 17%–27%).

Table 19: VSTAR3-1 and VSTAR3-2, multivariate search – frequency of superior loglikelihood

	VSTAR3-1				VSTAR3-2			
	Starting-value search							
	DE	SA	TA	GS	DE	SA	TA	GS
DE	-	29.30%	99.90%	89.90%	-	49.00%	100.00%	78.40%
SA	9.00%	-	99.90%	88.90%	4.90%	-	100.00%	72.90%
TA	0.00%	0.00%	-	17.00%	0.00%	0.00%	-	3.10%
GS	2.00%	3.10%	81.60%	-	5.80%	15.50%	96.70%	-
	ML estimation							
DE	-	25.90%	53.20%	49.90%	-	33.00%	42.70%	35.50%
SA	10.10%	-	51.80%	48.80%	8.50%	-	41.10%	32.80%
TA	18.90%	23.00%	-	25.40%	17.60%	25.20%	-	16.60%
GS	19.30%	22.90%	31.10%	-	17.90%	27.30%	28.30%	-

Remark for interpretation: Row better than column.
At least 0.05 per cent larger likelihood than other algorithm.

A conclusion from this is that the grid search with zoom has a tendency to find an inferior local optimum, which does not help the optimization algorithm to converge to global or at least to a superior local optimum. This argument receives support from the results in Table 20. Although the optimized loglikelihood is in more than 66% at least 0.01 per cent better than starting-value search, the final outcomes are still clearly worse than those of DE and SA. Hence, also the derivative-based algorithms does not help the GS to obtain better results. This could indicate finding a local, rather than a global, optimum.
SA and DE yield the best outcomes for final estimates with a slight edge for DE. From Table 21 it is seen that DE obtains the highest amount of superior outcomes both overall and across all procedures. The latter measure indicates the frequency of simulation runs for which an algorithm yields a 0.05 higher loglikelihood than all other algorithms.

Table 20: VSTAR1-3 and VSTAR2-3, comparison of starting-value search and optimization (loglikelihood)

	(i)		(ii)	
	VSTAR1-3	VSTAR2-3	VSTAR1-3	VSTAR2-3
DE	17.80%	29.70%	80.70%	68.10%
SA	20.40%	34.40%	78.90%	64.90%
TA	79.60%	88.00%	20.40%	12.00%
GS	66.60%	67.30%	30.20%	26.20%

(i) Optimized loglikelihood at least 0.01 per cent better than starting-value search.
(ii) Identical results of estimation and starting-value search (up to the fourth decimal).

Table 21: VSTAR3-1 and VSTAR3-2, multivariate search – best across all procedures

	(a)		(b)	
	VSTAR1-3	VSTAR2-3	VSTAR1-3	VSTAR2-3
DE	1290	1112	12.6%	11.1%
SA	1107	824	4.3%	3.4%
TA	673	594	6.9%	4.7%
GS	733	735	6.3%	4.4%

(a) total number of superior outcomes (better than at least one).
(b) frequency of superior outcomes across all procedures (better than all).

This may be due to the fact that DE, belonging to the class of population based methods, updates the whole set of potential solution simultaneously and is in that sense more successful than the other algorithms.

The results are reinforced by evaluating the magnitude of inferiority. Thereby, I assess by how much an algorithm is worse if it obtains already a lower loglikelihood value than another algorithm. Table 22 shows the mean and the 1% quantile of the (pairwise) differences of the loglikelihood value if an algorithm yields a worse loglikelihood than another one. The mean of the loglikelihood differences of the worse results is displayed in the upper part and the 1% quantile of the worse loglikelihood values is shown in the lower panel. The lower the values are, the worse the algorithm is. Ideally, the differences should be low as then the optimum found by the other algorithm is missed only closely. On average TA and GS with a zoom yield a lower likelihood than SA and DE, if they are already worse than SA and DE. The results of the 1% quantile of the worst results are most clear. If GS and TA are worse than DE and SA than they are clearly much worse than vice versa. Hence, GS and TA miss the optimum found by DE and SA by much more than it is the other way round.

The measure of inferiority indicates that SA and DE are preferable. Yet, one has to keep in mind that the number of likelihood evaluation of a grid with a zoom are much lower and it is much faster.

Table 22: VSTAR1-3 and VSTAR2-3, multivariate search search – magnitude of inferiority

	VSTAR3-1				VSTAR3-2			
	Mean of worse results							
	DE	SA	TA	GS	DE	SA	TA	GS
DE	-	−0.128	−0.448	−0.455	-	−0.122	−0.347	−0.292
SA	−0.168	-	−0.439	−0.451	−0.175	-	−0.351	−0.303
TA	−1.825	−1.843	-	−2.098	−2.347	−2.385	-	−3.010
GS	−1.487	−1.486	−1.656	-	−0.937	−0.955	−1.148	-
	1% quantile of worse results							
DE	-	−0.885	−3.786	−3.585	-	−0.634	−3.346	−2.868
SA	−1.155	-	−3.548	−3.242	−0.773	-	−2.742	−2.303
TA	−6.859	−6.871	-	−7.849	−13.017	−13.335	-	−15.353
GS	−8.253	−8.237	−9.750	-	−6.178	−6.959	−9.411	-

Remark for interpretation: Row is at least 0.05 per cent worse than column.
Mean and 1% quantile of the (pairwise) differences of the loglikelihood values if an algorithm yields already a 0.05 worse loglikelihood than another one.

5.2.4 Trivariate VSTAR model with single transition function

Finally, I describe the results of the VSTAR3-2 which is a trivariate Vector STAR model with zero restrictions and one transition variable governing the whole system. This makes a multivariate search procedure inevitable. For this type, I reduce the number of likelihood evaluations and use a normal (equation-by-equation) grid search implementation because only two parameter have to be optimized. The iterations of the heuristics are also decreased to the equation-by-equation set-up. For all algorithms a FGLS estimation instead of OLS due to the single transition function is applied in the starting-value search. When it comes to estimation, ML instead of NLS has to be used for the latter reason as well.

As can be seen from Table 23, SA and DE are particulary efficient. GS yields also convincing results but is slightly worse than DE and SA, whereas TA clearly ends up in the worst outcomes. In more than 50% the initial values obtained are worse than those of the other algorithms. This frequency is reduced after ML estimation, but still holds of considerable magnitude. Hence, GS, SA and DE seem to perform best with a slight edge for DE and SA.

Specifically, DE, SA and GS are already quite efficient in the starting value search as can be seen from Table 24. The optimized likelihood obtained after the estimation is only in around 1%–3% higher than that obtained after the starting-value search, whereas this frequency is clearly higher (67%) for TA. Although the outcomes of TA can be improved by an derivative-based algorithm, they still are not as good as those of the other algorithms. Hence, the preferable starting-value search methods for a higher-dimensional VSTAR process with one transition function requiring a multivariate search are SA and DE.

Table 23: VSTAR3-2, multivariate search search – frequency of superior loglikelihood

	Starting-value search				ML estimation			
	DE	SA	TA	GS	DE	SA	TA	GS
DE	-	0.00%	52.10%	1.70%	-	0.00%	12.80%	1.70%
SA	0.00%	-	51.80%	1.60%	0.00%	-	12.80%	1.60%
TA	0.00%	0.00%	-	0.30%	0.00%	0.00%	-	1.10%
GS	0.00%	0.00%	51.20%	-	0.00%	0.00%	12.40%	-

Remark for interpretation: Row better than column.
At least 0.05 per cent larger likelihood than other algorithm.

Table 24: VSTAR3-2, comparison of starting-value search and optimization (loglikelihood)

	(i)	(ii)
DE	1.00%	60.50%
SA	1.60%	39.00%
TA	67.50%	14.90%
GS	3.20%	50.00%

(i) Optimized loglikelihood at least 0.01 per cent better than starting-value search.
(ii) Identical results of estimation and starting-value search (up to the fourth decimal).

6 Conclusion

As long as the Vector STAR model has no cross-correlation across error terms no matter whether the process contains zero restrictions, the equation-by-equation starting-value search approach is preferable. Within this approach, no remarkable differences across procedures, namely, Grid Search (GS), Simulated Annealing (SA), Differential Evolution (DE) and Threshold Accepting (TA), arise. In the case of the heuristics (DE, SA, and TA), the starting-value search is only slightly improved by the derivative-based algorithm, indicating that they are already quite efficient in finding starting-values. Yet, GS benefits more from a classical optimization procedure. Once the VSTAR model becomes higher-dimensional, and thus more complex, the model should be estimated by ML. Moreover, an equation-by-equation approach with a derivative-based algorithm can handle such an optimization problem sufficiently well unless the errors are cross-correlated. The differences to the bivariate model are that i) the derivative-based algorithm clearly improves the outcome obtained by the starting-value search, and ii) the heuristic methods obtain slightly better results than the grid search. This may come from the more inflexible search space of a grid.

As soon as the error terms are cross-correlated, a multivariate starting-value search procedure should be used to achieve the highest likelihood value after the optimization. TA and the GS with a zoom do not yield convincing results compared to SA and DE. The latter two seem to be the best starting-value search methods for those VSTAR models for which a multivariate procedure is superior with a slight edge for DE. SA and DE

obtain also better loglikelihood values than the GS with a zoom and TA, assessing the magnitude of inferiority. If the latter methods are worse than the former, than they are much more worse. Yet, the GS with a zoom is based on a lower number of objective function evaluations than the other algorithms and is, consequently, faster. In a next step, one could assess whether the results of a grid with a "finer zoom" (higher number of grid points) still shows a tendency to find inferior optima rather than a global or at least superior local optimum.

If the VSTAR model is higher-dimensional and governed by one transition function, TA yields clearly the worst outcomes. SA and DE are slightly better than GS.

The following conclusions could be drawn for an empirical application. If the VSTAR model has a single transition function, SA and DE are preferable. However, if the transition function is equation-specific, one should initially check whether the errors of the VSTAR model are cross-correlated. This could be done statistically as well as by assessing whether correlation across equations are economically reasonable. If they are not correlated, it does not matter which method is applied unless the model has more than two dimensions. Then, one should rather prefer a heuristic method than the grid search. In an empirical application of a multivariate VSTAR model with cross-correlated errors, one should use DE or SA for finding starting-values.

Appendix

$$\mathbf{y_t} = \begin{bmatrix} .15 & .05 \\ .1 & .1 \end{bmatrix} \mathbf{y_{t-1}} + \begin{bmatrix} .1 & .1 \\ .05 & .15 \end{bmatrix} \mathbf{y_{t-2}} + \mathbb{G}\left(\begin{bmatrix} .05 & .15 \\ .1 & .1 \end{bmatrix} \mathbf{y_{t-1}} + \begin{bmatrix} .1 & .1 \\ .15 & .05 \end{bmatrix} \mathbf{y_{t-2}} \right) + \varepsilon_\mathbf{t} \quad (5)$$

$$\mathbf{y_t} = \begin{bmatrix} .05 & 0 \\ .05 & 0 \end{bmatrix} \mathbf{y_{t-1}} + \begin{bmatrix} 0 & .1 \\ .05 & .1 \end{bmatrix} \mathbf{y_{t-2}} + \begin{bmatrix} .05 & 0 \\ 0 & 0 \end{bmatrix} \mathbf{y_{t-3}} + \begin{bmatrix} 0 & 0 \\ .0 & .1 \end{bmatrix} \mathbf{y_{t-4}} + \begin{bmatrix} 0 & .15 \\ 0 & 0 \end{bmatrix} \mathbf{y_{t-5}} + \mathbb{G}\left(\begin{bmatrix} 0 & .15 \\ .15 & 0 \end{bmatrix} \mathbf{y_{t-1}} + \begin{bmatrix} .15 & .1 \\ 0 & .15 \end{bmatrix} \mathbf{y_{t-2}} + \begin{bmatrix} 0 & 0 \\ 0 & .1 \end{bmatrix} \mathbf{y_{t-3}} + \begin{bmatrix} .05 & 0 \\ .05 & 0 \end{bmatrix} \mathbf{y_{t-4}} \right) + \varepsilon_\mathbf{t} \quad (6)$$

$$\mathbf{y_t} = \begin{bmatrix} .12 & 0 & 0 \\ 0 & .31 & 0 \\ 0 & 0 & 0 \end{bmatrix} \mathbf{y_{t-1}} + \begin{bmatrix} 0 & -.2 & 0 \\ 0 & .02 & 0 \\ -.05 & 0 & .34 \end{bmatrix} \mathbf{y_{t-2}} + \begin{bmatrix} 0 & 0 & 0 \\ -.1 & 0 & .19 \\ 0 & 0 & 0 \end{bmatrix} \mathbf{y_{t-3}} + \begin{bmatrix} 0 & 0 & -.3 \\ 0 & 0 & 0 \\ 0 & 0 & .14 \end{bmatrix} \mathbf{y_{t-4}} + \begin{bmatrix} .4 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \mathbf{y_{t-6}} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -.28 \\ 0 & 0 & 0 \end{bmatrix} \mathbf{y_{t-9}} + \begin{bmatrix} 0 & .11 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & .09 \end{bmatrix} \mathbf{y_{t-12}} + \mathbb{G}\left(\begin{bmatrix} .18 & 0 & 0 \\ 0 & .21 & 0 \\ 0 & -.24 & .23 \end{bmatrix} \mathbf{y_{t-1}} + \begin{bmatrix} -.1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \mathbf{y_{t-2}} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -.4 \\ 0 & 0 & 0 \end{bmatrix} \mathbf{y_{t-3}} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & -.16 & 0 \end{bmatrix} \mathbf{y_{t-4}} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ .4 & 0 & 0 \end{bmatrix} \mathbf{y_{t-5}} + \begin{bmatrix} 0 & .22 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \mathbf{y_{t-6}} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & .21 \end{bmatrix} \mathbf{y_{t-7}} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & .55 \\ 0 & 0 & 0 \end{bmatrix} \mathbf{y_{t-11}} \right) + \varepsilon_\mathbf{t} \quad (7)$$

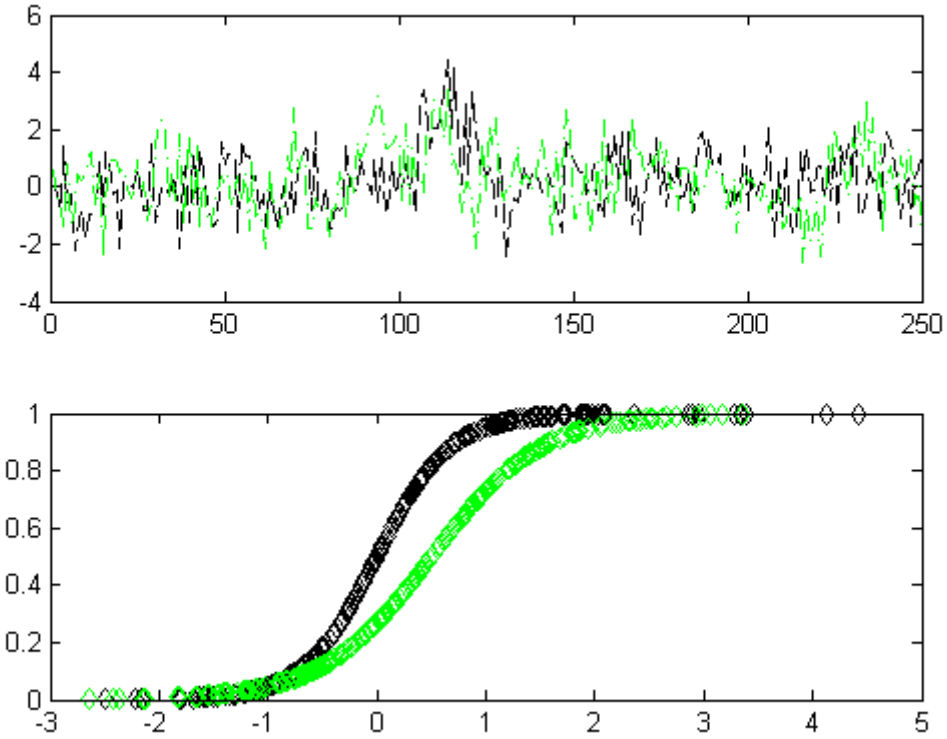

Figure 3: VSTAR1-1: bivariate VSTAR model without zero restrictions, moderate transition speed

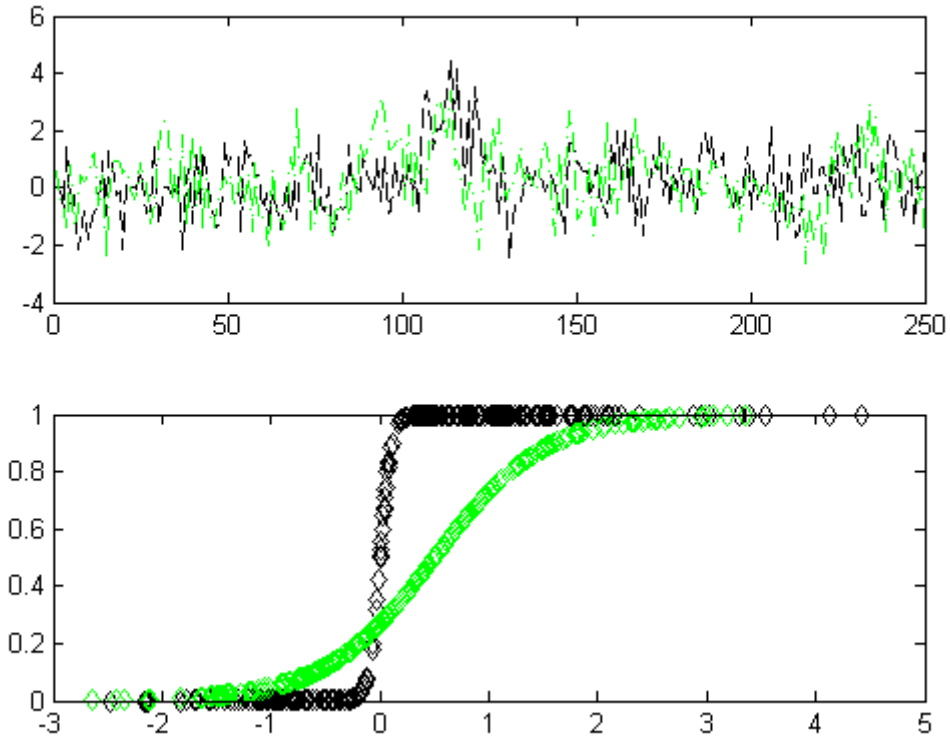

Figure 4: VSTAR1-2: bivariate VSTAR model without zero restrictions, moderate and high transition speed

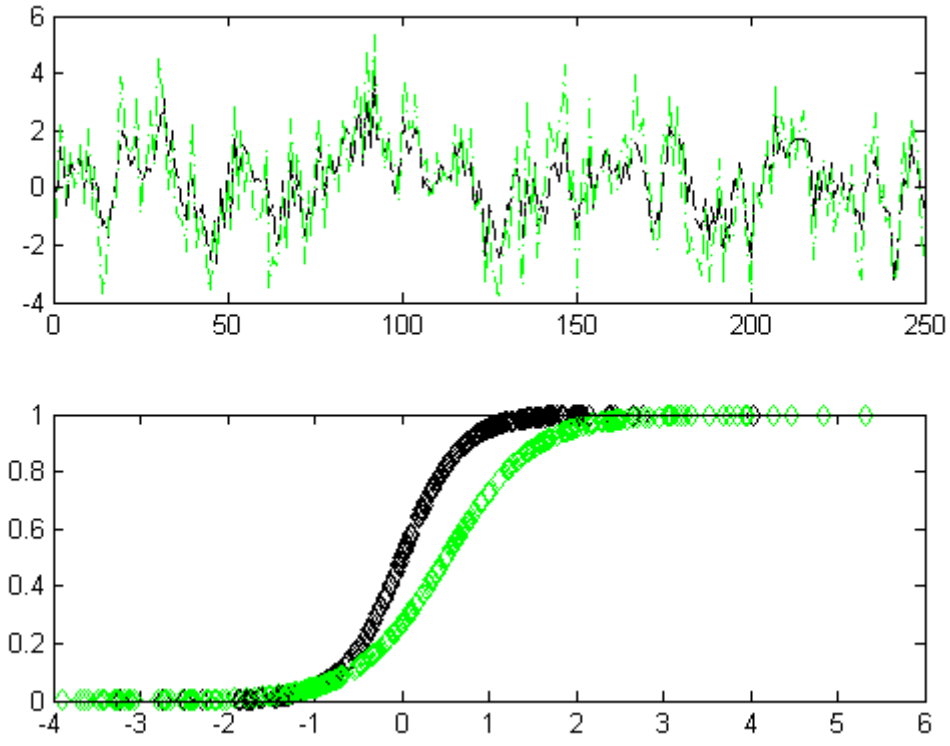

Figure 5: VSTAR1-3: bivariate VSTAR model without zero restrictions, cross-correlated errors

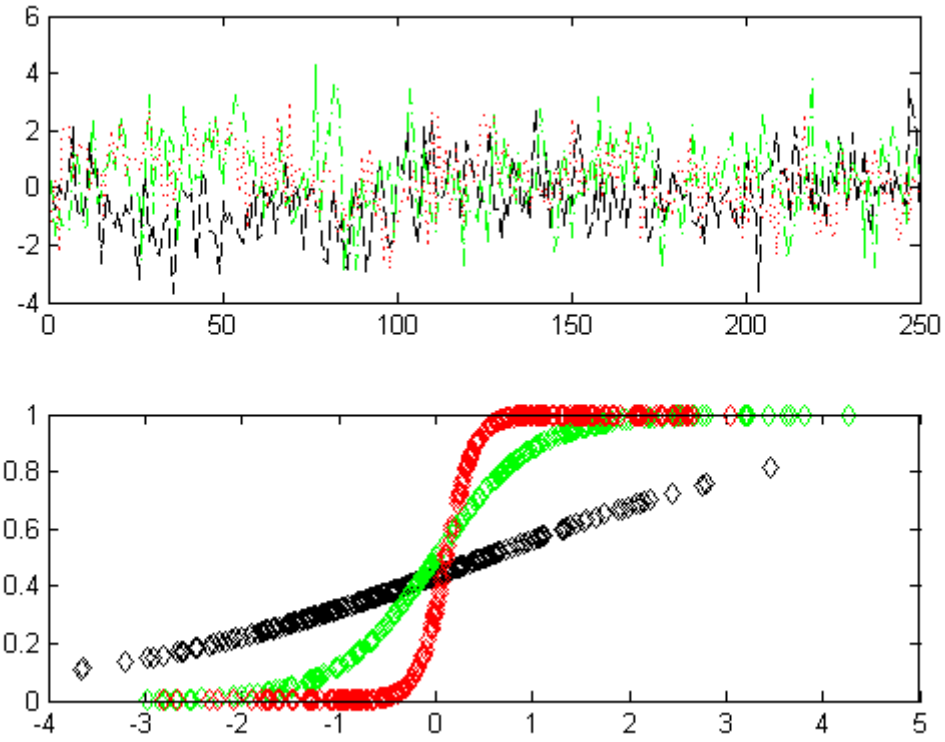

Figure 6: VSTAR3-1: trivariate VSTAR model with zero restrictions

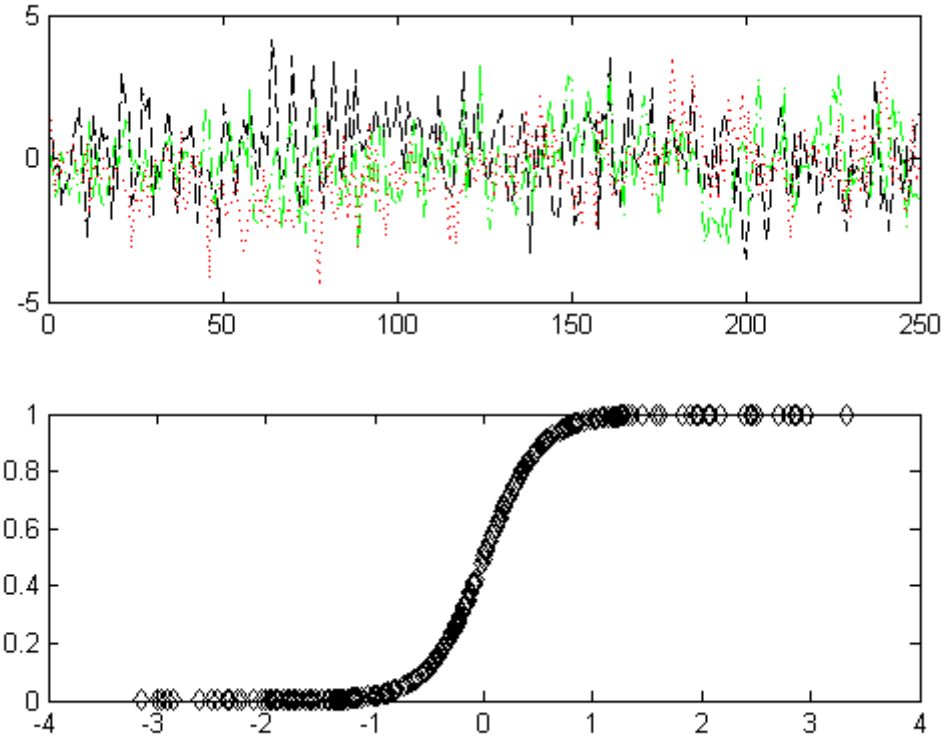

Figure 7: VSTAR3-2: trivariate VSTAR model with zero restrictions and single transition function

References

Andrews, D. W. K. (1993): "Tests for Parameter Instability and Structural Change with Unknown Change Point," *Econometrica*, 61(4), 821–56.

Baragona, R., F. Battaglia, and D. Cucina (2004): "Fitting piecewise linear threshold autoregressive models by means of genetic algorithms," *Computational Statistics & Data Analysis*, 47(2), 277–295.

Baragona, R., and D. Cucina (2013): "Multivariate Self-Exciting Threshold Autoregressive Modeling by Genetic Algorithms," *Journal of Economics and Statistics (Jahrbuecher fuer Nationaloekonomie und Statistik)*, 233(1), 3–21.

Battaglia, F., and M. K. Protopapas (2011): "Time-varying multi-regime models fitting by genetic algorithms," *Journal of Time Series Analysis*, 32(3), 237–252.

Caner, M., and B. E. Hansen (2001): "Threshold Autoregression with a Unit Root," *Econometrica*, 69(6), 1555–1596.

Chan, F., and M. McAleer (2002): "Maximum likelihood estimation of STAR and STAR-GARCH models: theory and Monte Carlo evidence," *Journal of Applied Econometrics*, 17(5), 509–534.

El-Shagi, M. (2011): "An evolutionary algorithm for the estimation of threshold vector error correction models," *International Economics and Economic Policy*, 8(4), 341–362.

Gilli, M., and P. Winker (2009): "Heuristic Optimization Methods in Econometrics," in *Handbook of Computational Econometrics*, ed. by D. Belsley, and E. Kontoghiorghes, pp. 81–119. John Wiley & Sons, Ltd.

Goffe, W. L., G. D. Ferrier, and J. Rogers (1994): "Global optimization of statistical functions with simulated annealing," *Journal of Econometrics*, 60(1-2), 65–99.

Gonzàlez, A., H. Rincon, and N. Rodriquez (2009): "The Transmission of Shocks to the Exchange Rate on the Inflation of imported goods in the presence of asymmetries," Working paper, Banco de la Repblica, Colombia.

Greene, W. H. (2003): *Econometric Analysis*. 5.ed. Pearson Eduction, Inc., New Jersey.

Hubrich, K., and T. Teräsvirta (2013): "Thresholds and Smooth Transitions in Vector Autoregressive Models," Research paper 2013-18, CREATES.

Leybourne, S., P. Newbold, and D. Vougas (1998): "Unit Roots and Smooth Transitions," *Journal of Time Series Analysis*, 19(1), 83 – 97.

Lyra, M., J. Paha, S. Paterlini, and P. Winker (2010): "Optimization heuristics for determining internal rating grading scales," *Computational Statistics & Data Analysis*, 54(11), 2693–2706.

Maringer, D., and M. Meyer (2008): "Smooth Transition Autoregressive Models – New Approaches to the Model Selection Problem," *Studies in Nonlinear Dynamics & Econometrics*, 12(1), 1 – 19.

Maringer, D., and P. Winker (2009): "The convergence of estimators based on heuristics: theory and application to a GARCH model," *Computational Statistics*, 24(3), 533–550.

Teräsvirta, T. (2004): "Smooth Transition Regression Modeling," in *Applied Time Series Econometrics*, ed. by H. Lütkepohl, and M. Krätzig. Cambridge University Press, Cambridge.

Teräsvirta, T., and Y. Yang (2013): "Specifcation, Estimation and Evaluation of Vector Smooth Transition Autoregressive Models with Applications," unpublished working paper, CREATES Research Paper.

van Dijk, D., T. Teräsvirta, and P. Franses (2002): "Smooth Transition Autoregressive Models - A Survey of Recent Developments," *Econometric Reviews*, 21(1), 1 – 47.

Winker, P., and K.-T. Fang (1997): "Application of Threshold-Accepting to the Evaluation of the Discrepancy of a Set of Points," *SIAM Journal on Numerical Analysis*, 34(5), pp. 2028–2042.

Wu, B., and C.-L. Chang (2002): "Using genetic algorithms to parameters (d,r) estimation for threshold autoregressive models," *Computational Statistics & Data Analysis*, 38(3), 315–330.

Yang, Z., Z. Tian, and Z. Yuan (2007): "GSA-based maximum likelihood estimation for threshold vector error correction model," *Computational Statistics & Data Analysis*, 52(1), 109–120.