



Article

# Analytical Gradients of Dynamic Conditional Correlation Models

Massimiliano Caporin <sup>1,\*</sup> , Riccardo (Jack) Lucchetti <sup>2</sup>  and Giulio Palomba <sup>2</sup><sup>1</sup> Department of Statistical Sciences, University of Padova, 35122 Padova PD, Italy<sup>2</sup> DISES, Università Politecnica delle Marche, 60121 Ancona AN, Italy; r.lucchetti@univpm.it (R.J.L.); g.palomba@univpm.it (G.P.)

\* Correspondence: massimiliano.caporin@unipd.it

Received: 31 January 2020; Accepted: 28 February 2020; Published: 4 March 2020



**Abstract:** We provide the analytical gradient of the full model likelihood for the Dynamic Conditional Correlation (DCC) specification by Engle (2002), the generalised version by Cappiello et al. (2006), and of the cDCC model by Aielli (2013). We discuss how the gradient might be further extended by introducing elements related to the conditional variance parameters, and discuss the issue arising from the estimation of constrained and/or reparametrised versions of the model. A computational simulation compares analytical versus numerical gradients, with a view to parameter estimation; we find that analytical differentiation yields more efficiency and improved accuracy.

**Keywords:** DCC; cDCC; GDCC; analytical gradient

## 1. Introduction

One of most challenging issues related to the class of Multivariate GARCH (MGARCH) models is the so-called curse of dimensionality: as a rule, the number of parameters of a MGARCH model diverges when the number of modelled variables increases. In general terms, denoting as  $n$  the number of variables, the number of parameters in a MGARCH model is  $O(n^b)$  with  $b > 1$ : in the VEC model by Engle and Kroner (1995)  $b = 4$ , while in the BEKK model, by the same authors,  $b = 2$ , and for the Dynamic Conditional Correlation (DCC) model by Engle (2002)  $b = 2$ . The solution most commonly adopted to overcome the curse of dimensionality is the introduction of constraints on the parameter space, leading to the so-called diagonal or scalar specifications, or to specifications driven by external information; see, among many others, Engle (2002); Bauwens et al. (2006); Caporin and McAleer (2008); and Caporin and Paruolo (2015). Few efforts have been posed on approaches which make it possible to ease those constraints and keep an MGARCH specification as close as possible to full parametrisation. Our approach goes into that direction, and is related to the work by Lucchetti (2002) on the BEKK model.

A key element for the feasibility of model estimation when the number of parameters is large is the availability of analytical gradients of the likelihood. In this work, we provide the analytical expression of the gradient for three well-known conditional correlation models: the Dynamic Conditional Correlation (DCC) model by Engle (2002), the generalisation proposed by Cappiello et al. (2006), and the more recent Consistent DCC (cDCC) model by Aielli (2013).

Notably, these models are generally estimated by two-step approaches: first, we recover conditional variance parameters by mean of univariate estimations; then, we maximise the conditional correlation likelihood with respect to correlation parameters, conditionally on the estimated variance parameters. In this way, the estimates clearly suffer from a loss of efficiency. However, when the correlation dynamic is highly restricted, for instance leading to scalar parametrisations, the number of parameters is quadratic in the number of variables, that is,  $b = 2$ , but the model estimation is

performed by a collection of univariate GARCH likelihoods (one for each marginal) and a correlation likelihood. The latter might take advantage of correlation targeting, thus estimating  $O(n^2)$  parameters via a sample statistic. Consequently, the model estimation becomes feasible irrespective of  $n$ .

While on the one side restricted parametrisations are feasible, on the other side they are extremely limited in terms of economic interpretation of the outcomes; in fact, they most likely neglect all possible interdependence among correlations and variances (i.e., they do not include spillover effects). The introduction of the gradient for the most general DCC specification will allow estimating the fully parametrised specifications when the cross-sectional dimension ( $n$ ) is not too large, and conditionally to the computing power which is available to the researcher. Alternatively, by adapting the gradient to intermediate specifications might allow reintroducing even limited spillovers within and between correlation and/or variance dynamic.

Our contribution could be potentially considered as a tool for measuring the efficiency loss of the two-step estimation approaches. Moreover, it could be applied in the implementation of full likelihood estimation approaches, and it will be mostly useful for the application of the maximisation by parts estimation methodology as suggested in [Fan et al. \(2015\)](#). However, the complexity of the following analytical formulae represents a serious challenge for the previous purposes.

Other studies have already tackled the problem of deriving the analytical representation of the DCC model gradient—[Engle and Sheppard \(2001\)](#) presents a partial gradient without taking into account its recursive structure, while [Hafner and Herwartz \(2008\)](#) report the gradient of the second stage likelihood of the DCC model of [Engle \(2002\)](#), without considering the derivatives with respect to the first stage parameters. We also mention other works reporting analytical forms for different multivariate GARCH models—[Ling and McAleer \(2003\)](#) focused on their VARMA-GARCH model that nests the constant conditional correlation model of [Bollerslev \(1990\)](#), and [Lucchetti \(2002\)](#) provides the analytical expression for the gradient of the BEKK model of [Engle and Kroner \(1995\)](#). We complement this strand of the financial econometrics literature with respect to the most common conditional correlation models.

Beside the analytical derivatives, and a discussion on how they could be adapted to the introduction of constraints on the model parameters, we also provide empirical evidences supporting the use of analytical score. A simulation experiment shows that the use of the analytical gradient has clear advantages as compared to the numerical gradient, with a sensible reduction in the CPU time, unless in the case of extremely restricted specifications where its impact is detrimental. However, even in these cases analytical differentiation yields vastly improved accuracy.

The paper proceeds as follows. In the next section we provide preliminary elements for the following derivation of the analytical gradient. Section 3 reports several differentials which are then used in Section 4 to obtain the gradient of the DCC model of [Engle \(2002\)](#). Section 4.2 contains the gradient of the cDCC model of [Aielli \(2013\)](#) and of the GDCC model of ([Cappiello et al. 2006](#)). Sections 5 and 6 are devoted to a detailed analysis of the issues that arise when actually building a software implementation of the gradient, together with a simulation in Section 7. Section 8 concludes.

## 2. Preliminaries

In this section, we establish the notation we will use in the rest of the paper and define explicitly the class of models we will be working on.

### 2.1. Notation

In the following sections we make use of the following symbols, operators and matrix relations; whenever possible, we adhere to the conventions spelled out in [Magnus and Neudecker \(1999\)](#):

- $\mathbf{1}_n$  and  $\mathbf{0}_n$  are  $n$ -dimensional vectors whose elements are all 1 and 0, respectively;
- $I_n$  is the identity matrix of dimension  $n$ ;
- for a diagonal matrix  $A$  we denote by  $A^{1/2}$  the square root of the matrix, where the square root operates element-wise;

- $\text{diag}(a)$  is a diagonal matrix with the vector  $a$  on its main diagonal;
- $\text{dg}(A)$  creates a vector from the diagonal of matrix  $A$ ;
- the  $\delta[A]$  operator is defined as  $\delta[A] \equiv \text{diag}[\text{vec}(A)]$ ;
- the “:” symbol denotes horizontal concatenation of matrices;
- the operator  $\text{vech}$  stacks the lower triangular part of a matrix, while the operator  $\text{vechl}$  stacks the lower triangular part of a matrix excluding the diagonal;
- for a square diagonal matrix  $A$  of dimension  $n$  with the vector  $a$  over the main diagonal, we denote by  $G_n$  the  $n^2 \times n$  matrix satisfying  $\text{vec}(A) = G_n a$ ;
- for a square matrix  $A$  of dimension  $n$  we denote by  $K_{nn}$  the  $n^2 \times n^2$  commutation matrix satisfying  $\text{vec}(A') = K_{nn} \text{vec}(A)$ ;
- we define the “symmetrisation” matrix as  $N_n = \frac{1}{2}(I_{n^2} + K_{nn})$ , which is symmetric and idempotent with rank  $\frac{n(n+1)}{2}$ ;
- for a symmetric matrix  $A$ , we denote by  $D_n$  the  $n^2 \times \frac{n(n+1)}{2}$  duplication matrix satisfying  $D_n \text{vech}(A) = \text{vec}(A)$ ;
- $L_n$  is the  $\frac{n(n+1)}{2} \times n^2$  “elimination matrix”, satisfying  $L_n \text{vec}(A) = \text{vech}(A)$ ;
- for a square matrix  $A$  we denote by  $\Delta_n$  the  $n^2 \times n^2$  selection matrix satisfying  $\Delta_n \text{vec}(A) = \text{vec}(\text{diag}(\text{dg}(A)))$ . This matrix eliminates the off-diagonal elements in the vectorisation of  $A$ ;
- $dA$  denotes the differential of matrix  $A$ ;
- the symbol  $\odot$  is used for the Hadamard (element-by-element) matrix product;
- a property we will use repeatedly is:  $\text{vec}(A \odot B) = \delta[A] \text{vec}(B)$  where  $A$  and  $B$  are square matrices of the same order; therefore,  $\text{vec}(dB \odot A) = \text{vec}(A \odot dB) = \delta[A] \text{vec}(dB)$ ;
- if  $R$  is a correlation matrix, we denote by  $S_n$  the duplication matrix of dimension  $n^2 \times \frac{n(n-1)}{2}$  satisfying  $\text{vec}(R) = \text{vec}(I_n) + S_n \text{vechl}(R)$ ; then,  $\text{vec}(dR) = S_n \text{vechl}(dR)$ .

For selected and relevant matrices we report below them and within parenthesis their dimensions. For general matrix properties and matrix differentials see Magnus and Neudecker (1999) and Lütkepohl (1996).

## 2.2. Dynamic Conditional Correlation Models

In the following we focus on three dynamic conditional correlation models. The first, our reference model, is the DCC model of Engle (2002). Denote an  $n$ -dimensional vector of asset returns by  $x_t$  and the information set at time  $t - 1$  by  $\mathcal{I}_{t-1}$ . Since our object of interest here are covariances and correlations, we consider the mean-centred returns  $\varepsilon_t = x_t - E[x_t | \mathcal{I}_{t-1}]$  and we assume they follow some unspecified conditional density satisfying

$$\varepsilon_t | \mathcal{I}_{t-1} \sim D(\mathbf{0}_n, \Sigma_t).$$

The conditional covariance matrix can be split into

$$\Sigma_t = V_t^{1/2} R_t V_t^{1/2},$$

where  $V_t = \text{diag}(v_t)$  is the diagonal matrix of conditional variances, where  $v_t = [\sigma_{1,t}^2, \sigma_{2,t}^2, \dots, \sigma_{n,t}^2]$ , and  $R_t$  is a dynamic correlation matrix. The conditional variances are governed by the following dynamic equation

$$v_t = \omega + \mathcal{A}(\varepsilon_{t-1} \odot \varepsilon_{t-1}) + \mathcal{B}v_{t-1},$$

where  $\omega$  is an  $n$ -dimensional parameter vector while  $\mathcal{A}$  and  $\mathcal{B}$  are square parameter matrices of dimension  $n$ . Note that the adopted specification nests as a special case a set of univariate GARCH(1,1) for all assets (under a diagonality assumption for the parameter matrices  $\mathcal{A}$  and  $\mathcal{B}$ ). Note this structure

has already been used in [Ling and McAleer \(2003\)](#). The dynamic correlation matrix is further decomposed into

$$R_t = \left( \tilde{Q}_t^{-1/2} Q_t \tilde{Q}_t^{-1/2} \right),$$

where  $\tilde{Q}_t = \text{dg}(Q_t)$  and

$$Q_t = \Gamma + A \odot (\eta_{t-1} \eta'_{t-1} - \Gamma) + B \odot (Q_{t-1} - \Gamma), \tag{1}$$

in which  $\eta_t = V_t^{-1/2} \varepsilon_t$ ,  $A$  and  $B$  are symmetric parameter matrices and  $\Gamma$  is a correlation matrix. We consider here the most general DCC specification. Restricted forms, such as the scalar one, could be obtained as special cases by appropriate restrictions on the matrices  $A$  and  $B$ . For instance, the scalar model is given by  $A = \alpha \cdot \iota_n \iota'_n$  and  $B = \beta \cdot \iota_n \iota'_n$ , where  $\alpha$  and  $\beta$  are scalar parameters. See Section 6 for a fuller discussion.

The second model is the cDCC model by [Aielli \(2013\)](#), that solves some drawbacks of the DCC model of [Engle \(2002\)](#). The two models are very similar. In fact, the cDCC only replaces Equation (1) with

$$Q_t = \Gamma + A \odot \left( \tilde{Q}_{t-1}^{1/2} \eta_{t-1} \eta'_{t-1} \tilde{Q}_{t-1}^{1/2} - \Gamma \right) + B \odot (Q_{t-1} - \Gamma). \tag{2}$$

The interested reader should refer to [Aielli \(2013\)](#) for additional details on model properties and for the comparison between DCC and cDCC. For the cDCC specification  $A$  and  $B$  are again two symmetric parameter matrices. Differently from the DCC of [Engle \(2002\)](#),  $\Gamma$  is a symmetric matrix with ones over the main diagonal (not necessarily positive definite).

The third model is the Generalised DCC (GDCC) of [Cappiello et al. \(2006\)](#) that considers a quadratic structure in Equation (1)

$$Q_t = \Gamma - A \Gamma A' - B \Gamma B' + A \eta_{t-1} \eta'_{t-1} A' + B Q_{t-1} B',$$

where the parameter matrices are not required to be symmetric,  $\Gamma$  is a correlation matrix and the intercept  $\Gamma - A \Gamma A' - B \Gamma B'$  must be positive definite.

We stress that we choose a very simple dynamic for the conditional variances only for illustrative purposes. More flexible model structures can be used, even with the adoption of different GARCH-type dynamics across the assets. In the following, we provide the gradient for our conditional variance specification, and discuss in more details the implementation and computational aspects of the gradient only referring to the conditional correlation parameters. Such a choice depends on the fact that different specifications for the conditional variance dynamic might lead to different implementation and computational strategies for the full model analytical gradient. We will further discuss this issue in Section 5.

### 3. Differentials

We report here a set of differentials which will be then used to construct the gradient of DCC models. For simplicity, we focus on the Normal likelihood but our approach can be generalised to other densities. Alternatively, the following results might be anyway used within a Quasi Maximum Likelihood estimation approach. Given a sample of  $T$  observations for  $n$  time series, the full model likelihood is given by

$$L_T(\theta) = \sum_{t=1}^T l_t(\theta),$$

where  $\theta$  is the parameter vector of the full likelihood, that is

$$\theta = \left[ \omega' : \text{vec}(A)' : \text{vec}(B)' : \text{vech}(\Gamma)' : \text{vech}(A)' : \text{vech}(B)' \right]'$$

Note that  $\theta$  has dimension  $d = n + n^2 + n^2 + \frac{n(n-1)}{2} + \frac{n(n+1)}{2} + \frac{n(n+1)}{2}$ , also written as  $d = d_1 + d_2 + d_3 + d_4 + d_5 + d_6$ . In order to keep notation clean, however, we will omit the explicit dependence of the key quantities from  $\theta$  and write  $l_t$  instead of  $l_t(\theta)$  and so on for similar cases.

After dropping constant terms, we have

$$l_t \approx -\frac{1}{2} \log |\Sigma_t| - \frac{1}{2} \varepsilon_t' \Sigma_t^{-1} \varepsilon_t. \tag{3}$$

### 3.1. Differential of the Full Likelihood

The differential for the likelihood  $l_t$  is given by

$$dl_t = -\varepsilon_t' \Sigma_t^{-1} d\varepsilon_t - \frac{1}{2} d \log |\Sigma_t| + \frac{1}{2} \varepsilon_t' \Sigma_t^{-1} (d\Sigma_t) \Sigma_t^{-1} \varepsilon_t. \tag{4}$$

We now assume that  $d\varepsilon_t = \mathbf{0}_n$  given that we focus on mean residuals and do not take into account possible mean parameters.

By the properties of the trace and vec operators we have

$$d \log |\Sigma_t| = \text{tr} \left( \Sigma_t^{-1} d\Sigma_t \right) = \text{vec} \left( \Sigma_t^{-1} \right)' \text{vec} (d\Sigma_t) \tag{5}$$

and

$$\varepsilon_t' \Sigma_t^{-1} (d\Sigma_t) \Sigma_t^{-1} \varepsilon_t = \left( \varepsilon_t' \Sigma_t^{-1} \otimes \varepsilon_t' \Sigma_t^{-1} \right) \text{vec} (d\Sigma_t). \tag{6}$$

Replace (5) and (6) into (4) to obtain

$$dl_t = \frac{1}{2} \left( \varepsilon_t' \Sigma_t^{-1} \otimes \varepsilon_t' \Sigma_t^{-1} - \text{vec} \left( \Sigma_t^{-1} \right)' \right) \text{vec} (d\Sigma_t). \tag{7}$$

We do not report the differential with respect to the non-duplicate elements of  $\Sigma_t$  to get a more direct relation with properties of the vec operator and given the model structure.

### 3.2. Differential of the Conditional Covariance Matrix

We now report the differential of the conditional covariance matrix taking into account its multiplicative structure:

$$d\Sigma_t = d \left( V_t^{1/2} R_t V_t^{1/2} \right) = V_t^{1/2} R_t dV_t^{1/2} + dV_t^{1/2} R_t V_t^{1/2} + V_t^{1/2} dR_t V_t^{1/2}. \tag{8}$$

We then consider the vec representation of the differential

$$\text{vec} (d\Sigma_t) = \text{vec} \left( V_t^{1/2} R_t dV_t^{1/2} \right) + \text{vec} \left( dV_t^{1/2} R_t V_t^{1/2} \right) + \text{vec} \left( V_t^{1/2} dR_t V_t^{1/2} \right)$$

and, by properties of the vec operator, we have

$$\text{vec} \left( V_t^{1/2} dR_t V_t^{1/2} \right) = \left( V_t^{1/2} \otimes V_t^{1/2} \right) \text{vec} (dR_t),$$

$$\text{vec} \left( dV_t^{1/2} R_t V_t^{1/2} \right) = \left( I_n \otimes V_t^{1/2} R_t \right) \text{vec} \left( dV_t^{1/2} \right),$$

and

$$\text{vec} \left( dV_t^{1/2} R_t V_t^{1/2} \right) = K_{nn} \text{vec} \left( V_t^{1/2} R_t dV_t^{1/2} \right).$$

For the diagonal matrix of conditional standard deviations  $V_t^{1/2}$  the following holds

$$\text{vec} \left( dV_t^{1/2} \right) = \frac{1}{2} \left( I_n \otimes V_t^{-1/2} \right) \text{vec} (dV_t),$$

where  $V_t$  is the diagonal matrix of conditional variances.

Then, using the diagonality of  $V_t$  we have

$$\text{vec}(dV_t) = G_n dv_t,$$

where  $v_t$  is the vector of conditional variances. After collecting terms we have

$$\text{vec}(d\Sigma_t) = N_n \left[ \left( I_n \otimes V_t^{1/2} R_t V_t^{-1/2} \right) G_n dv_t + \left( V_t^{1/2} \otimes V_t^{1/2} \right) \text{vec}(dR_t) \right]. \tag{9}$$

Putting together Equations (7) and (9), the differential of the full likelihood becomes

$$dl_t = \frac{1}{2} \left[ \varepsilon_t' \Sigma_t^{-1} \otimes \varepsilon_t' \Sigma_t^{-1} - \text{vec}(\Sigma_t^{-1})' \right] \times \\ \times N_n \left[ \left( I_n \otimes V_t^{1/2} R_t V_t^{-1/2} \right) G_n dv_t + \left( V_t^{1/2} \otimes V_t^{1/2} \right) \text{vec}(dR_t) \right];$$

note, however, that the Jacobian term in Equation (7) is the vectorisation of a symmetric matrix, and therefore remains unchanged if multiplied by  $N_n$ . The expression above, therefore, becomes

$$dl_t = \frac{1}{2} \left[ \varepsilon_t' \Sigma_t^{-1} \otimes \varepsilon_t' \Sigma_t^{-1} - \text{vec}(\Sigma_t^{-1})' \right] \times \\ \times \left[ \left( I_n \otimes V_t^{1/2} R_t V_t^{-1/2} \right) G_n dv_t + \left( V_t^{1/2} \otimes V_t^{1/2} \right) \text{vec}(dR_t) \right]$$

which can be abbreviated to

$$dl_t = C_{1,t} dv_t + C_{2,t} dR_t. \tag{10}$$

The calculation of the two partial Jacobian terms  $C_{1,t}$  and  $C_{2,t}$  is greatly simplified by defining

$$\zeta_t \equiv V_t^{1/2} \Sigma_t^{-1} \varepsilon_t. \tag{11}$$

The first partial Jacobian  $C_{1,t}$  becomes

$$C_{1,t} = \frac{1}{2} \left[ \varepsilon_t' \Sigma_t^{-1} \otimes \varepsilon_t' \Sigma_t^{-1} - \text{vec}(\Sigma_t^{-1})' \right] \left( I_n \otimes V_t^{1/2} R_t V_t^{-1/2} \right) G_n \\ = \frac{1}{2} \left[ \left( \varepsilon_t' \Sigma_t^{-1} \otimes \zeta_t' R_t V_t^{-1/2} \right) - \text{vec}(V_t^{-1})' \right] G_n \tag{12}$$

since

$$\left( I_n \otimes V_t^{-1/2} R_t V_t^{1/2} \right) \text{vec}(\Sigma_t^{-1}) = \text{vec}(V_t^{-1/2} R_t V_t^{1/2} \Sigma_t^{-1}) = \text{vec}(V_t^{-1}).$$

As for the second one,

$$C_{2,t} = \frac{1}{2} \left[ \zeta_t' \otimes \zeta_t' - \text{vec}(R_t^{-1})' \right] \tag{13}$$

where the equality

$$\text{vec}(\Sigma_t^{-1})' \left( V_t^{1/2} \otimes V_t^{1/2} \right) = \text{vec}(V_t^{1/2} \Sigma_t^{-1} V_t^{1/2})' \\ = \text{vec} \left[ \left( V_t^{-1/2} \Sigma_t V_t^{-1/2} \right)^{-1} \right]' \\ = \text{vec}(R_t^{-1})'$$

was used.

### 3.3. Differential of the Vector of Conditional Variances

Now focus on the  $dv_t$  term, keeping in mind we adopt a simplified parametrisation where all conditional variances follow a GARCH(1,1)-type dynamic, the one adopted in [Ling and McAleer \(2003\)](#). Recalling that  $v_t = \omega + \mathcal{A}(\varepsilon_{t-1} \odot \varepsilon_{t-1}) + \mathcal{B}v_{t-1}$ , we have the relation

$$dv_t = d\omega + d\mathcal{A}(\varepsilon_{t-1} \odot \varepsilon_{t-1}) + 2\mathcal{A}(\varepsilon_{t-1} \odot d\varepsilon_{t-1}) + d\mathcal{B}v_{t-1} + \mathcal{B}dv_{t-1}.$$

Given that  $d\varepsilon_{t-1} = \mathbf{0}_n$  this simplifies into

$$dv_t = d\omega + d\mathcal{A}(\varepsilon_{t-1} \odot \varepsilon_{t-1}) + d\mathcal{B}v_{t-1} + \mathcal{B}dv_{t-1},$$

and then by standard differentiation rules we have

$$dv_t = d\omega + \left[ I_n \otimes (\varepsilon_{t-1} \odot \varepsilon_{t-1})' \right] \text{vec}(d\mathcal{A}') + (I_n \otimes v'_{t-1}) \text{vec}(d\mathcal{B}') + \mathcal{B}dv_{t-1}.$$

We consider the differential with respect to the  $\text{vec}$  of parameter matrices (removing thus the transpose), obtaining

$$\begin{aligned} dv_t &= d\omega + \left[ I_n \otimes (\varepsilon_{t-1} \odot \varepsilon_{t-1})' \right] K_{nn} \text{vec}(d\mathcal{A}) + \\ &\quad + (I_n \otimes v'_{t-1}) K_{nn} \text{vec}(d\mathcal{B}) + \mathcal{B}dv_{t-1} \\ &= d\omega + W_{1,t} \text{vec}(d\mathcal{A}) + W_{2,t} \text{vec}(d\mathcal{B}) + \mathcal{B}dv_{t-1}, \end{aligned} \tag{14}$$

where

$$W_{1,t} = \left[ I_n \otimes (\varepsilon_{t-1} \odot \varepsilon_{t-1})' \right] K_{nn} \tag{15}$$

$(n \times n^2)$

and

$$W_{2,t} = (I_n \otimes v'_{t-1}) K_{nn}. \tag{16}$$

$(n \times n^2)$

Note that the matrices  $\mathcal{A}$  and  $\mathcal{B}$  need not be symmetric, but in many cases they may have some kind of structure so that the differential with respect to more restricted specifications could be easily obtained. For instance, a collection of univariate GARCH(1,1) corresponds to diagonal parameter matrices, and thus to  $\text{vec}(d\mathcal{A}) = G_n da$  where  $a$  is an  $n$ -dimensional parameter vector (similarly for the matrix  $\mathcal{B}$ ).

### 3.4. Differential of the Conditional Correlation Matrix

We now focus on  $\text{vec}(dR_t)$ . Starting from the decomposition of the correlation matrix we have

$$\begin{aligned} dR_t &= d\left(\tilde{Q}_t^{-1/2} Q_t \tilde{Q}_t^{-1/2}\right) \\ &= d\tilde{Q}_t^{-1/2} Q_t \tilde{Q}_t^{-1/2} + \tilde{Q}_t^{-1/2} dQ_t \tilde{Q}_t^{-1/2} + \tilde{Q}_t^{-1/2} Q_t d\tilde{Q}_t^{-1/2}. \end{aligned}$$

The diagonal matrix  $\tilde{Q}_t^{-1/2}$  is a function of elements of  $Q_t$ , we thus report first its differential with respect to  $Q_t$

$$d\tilde{Q}_t^{-1/2} = d\left[\left(\tilde{Q}_t^{1/2}\right)^{-1}\right] = -\tilde{Q}_t^{-1/2} d\tilde{Q}_t^{1/2} \tilde{Q}_t^{-1/2}$$

which follows by properties of differentials. Then

$$d\tilde{Q}_t^{1/2} = \frac{1}{2} \tilde{Q}_t^{-1/2} d\tilde{Q}_t.$$

By substituting these results into the differential of the conditional correlation matrix we have

$$dR_t = -\frac{1}{2}\tilde{Q}_t^{-1}d\tilde{Q}_tR_t + \tilde{Q}_t^{-1/2}dQ_t\tilde{Q}_t^{-1/2} - \frac{1}{2}R_t d\tilde{Q}_t\tilde{Q}_t^{-1}$$

where we used  $d\tilde{Q}_t^{1/2} = \tilde{Q}_t^{-1/2}d\tilde{Q}_t = d\tilde{Q}_t\tilde{Q}_t^{-1/2}$ , which comes from the diagonality (hence, commutability) of  $\tilde{Q}_t$  and  $\tilde{Q}_t^{-1/2}$ .

Turning to the vec of the correlation matrix, we have

$$\begin{aligned} \text{vec}(dR_t) &= -\frac{1}{2}\left(R_t \otimes \tilde{Q}_t^{-1}\right)\text{vec}(d\tilde{Q}_t) - \frac{1}{2}\left(\tilde{Q}_t^{-1} \otimes R_t\right)\text{vec}(d\tilde{Q}_t) + \\ &\quad + \left(\tilde{Q}_t^{-1/2} \otimes \tilde{Q}_t^{-1/2}\right)\text{vec}(dQ_t). \end{aligned}$$

Note that  $\text{vec}(d\tilde{Q}_t)$  is the vectorisation of a symmetric matrix, so  $\text{vec}(d\tilde{Q}_t) = K_{nn}\text{vec}(d\tilde{Q}_t)$ . As a consequence,

$$\begin{aligned} \left(\tilde{Q}_t^{-1} \otimes R_t\right)\text{vec}(d\tilde{Q}_t) &= \left(\tilde{Q}_t^{-1} \otimes R_t\right)K_{nn}\text{vec}(d\tilde{Q}_t) \\ &= K_{nn}\left(R_t \otimes \tilde{Q}_t^{-1}\right)\text{vec}(d\tilde{Q}_t) \end{aligned}$$

and therefore

$$-\frac{1}{2}\left[\left(R_t \otimes \tilde{Q}_t^{-1}\right) + \left(\tilde{Q}_t^{-1} \otimes R_t\right)\right]\text{vec}(d\tilde{Q}_t) = -N_n\left(R_t \otimes \tilde{Q}_t^{-1}\right)\text{vec}(d\tilde{Q}_t).$$

Finally, by using  $\text{vec}(d\tilde{Q}_t) = \Delta_n\text{vec}(dQ_t)$ , we obtain

$$\text{vec}(dR_t) = C_{3,t}\text{vec}(dQ_t),$$

where

$$C_{3,t} = \left[ \begin{matrix} (\tilde{Q}_t^{-1/2} \otimes \tilde{Q}_t^{-1/2}) - N_n (R_t \otimes \tilde{Q}_t^{-1}) \Delta_n \end{matrix} \right]_{(n^2 \times n^2)}. \tag{17}$$

### 3.5. Differential of the Dynamic Matrix $Q_T$

Note that the dynamic of the DCC model is given by

$$Q_t = \Gamma + A \odot (\eta_{t-1}\eta'_{t-1} - \Gamma) + B \odot (Q_{t-1} - \Gamma).$$

The differential is thus

$$\begin{aligned} dQ_t &= (\iota_n \iota'_n - A - B) \odot d\Gamma + dA \odot (\eta_{t-1}\eta'_{t-1} - \Gamma) + dB \odot (Q_{t-1} - \Gamma) + \\ &\quad + B \odot dQ_{t-1} + A \odot d(\eta_{t-1}\eta'_{t-1}). \end{aligned}$$

For the last term we can write

$$\begin{aligned} d(\eta_{t-1}\eta'_{t-1}) &= d\left(V_{t-1}^{-1/2}\varepsilon_{t-1}\varepsilon'_{t-1}V_{t-1}^{-1/2}\right) \\ &= dV_{t-1}^{-1/2}\varepsilon_{t-1}\varepsilon'_{t-1}V_{t-1}^{-1/2} + V_{t-1}^{-1/2}\varepsilon_{t-1}\varepsilon'_{t-1}dV_{t-1}^{-1/2}, \end{aligned}$$



where two differentials have been dropped since here we are assuming for the sake of simplicity<sup>1</sup> that  $d\varepsilon_t = \mathbf{0}_n$  for all values of  $t$ . Using previous results on the differentials of the conditional standard deviations we have

$$dV_{t-1}^{-1/2} = -\frac{1}{2}V_{t-1}^{-1/2}dV_{t-1}V_{t-1}^{-1}$$

and using this last result we have

$$\begin{aligned} d(\eta_{t-1}\eta'_{t-1}) &= -V_{t-1}^{-1}dV_{t-1}V_{t-1}^{-1/2}\varepsilon_{t-1}\varepsilon'_{t-1}V_{t-1}^{-1/2} - V_{t-1}^{-1/2}\varepsilon_{t-1}\varepsilon'_{t-1}V_{t-1}^{-1/2}dV_{t-1}V_{t-1}^{-1} \\ &= -\frac{1}{2}V_{t-1}^{-1}dV_{t-1}\eta_{t-1}\eta'_{t-1} - \frac{1}{2}\eta_{t-1}\eta'_{t-1}dV_{t-1}V_{t-1}^{-1}. \end{aligned}$$

Taking the vec of the previous expression we obtain

$$\begin{aligned} \text{vec}(d(\eta_{t-1}\eta'_{t-1})) &= -\frac{1}{2}\left[(\eta_{t-1}\eta'_{t-1}) \otimes V_{t-1}^{-1}\right] \text{vec}(dV_{t-1}) + \\ &\quad -\frac{1}{2}\left[V_{t-1}^{-1} \otimes (\eta_{t-1}\eta'_{t-1})\right] \text{vec}(dV_{t-1}). \end{aligned}$$

By a reasoning similar to the one in the previous subsection, we note that  $V_{t-1}$  is diagonal, hence symmetric and therefore  $K_{nn}\text{vec}(dV_{t-1}) = \text{vec}(dV_{t-1})$ . As a consequence,

$$\begin{aligned} \text{vec}(d(\eta_{t-1}\eta'_{t-1})) &= -\frac{1}{2}\left[(\eta_{t-1}\eta'_{t-1}) \otimes V_{t-1}^{-1}\right] \text{vec}(dV_{t-1}) + \\ &\quad -\frac{1}{2}\left[V_{t-1}^{-1} \otimes (\eta_{t-1}\eta'_{t-1})\right] K_{nn}\text{vec}(dV_{t-1}) \\ &= -N_n\left[V_{t-1}^{-1} \otimes (\eta_{t-1}\eta'_{t-1})\right] G_n dv_{t-1}. \end{aligned} \tag{18}$$

For other elements in  $\text{vec}(dQ_t)$  results follow from properties of differentials and matrix operators. For the  $\Gamma$  matrix we have

$$\begin{aligned} \text{vec}((\iota_n \iota'_n - A - B) \odot d\Gamma) &= \delta[\iota_n \iota'_n - A - B] \text{vec}(d\Gamma) \\ &= \delta[\iota_n \iota'_n - A - B] S_n \text{vech}(d\Gamma). \end{aligned}$$

For the two parameter matrices driving the dynamics we obtain, instead,

$$\begin{aligned} \text{vec}(dA \odot (\eta_{t-1}\eta'_{t-1} - \Gamma)) &= \text{vec}((\eta_{t-1}\eta'_{t-1} - \Gamma) \odot dA) \\ &= \delta[\eta_{t-1}\eta'_{t-1} - \Gamma] \text{vec}(dA) \\ &= \delta[\eta_{t-1}\eta'_{t-1} - \Gamma] D_n \text{vech}(dA), \end{aligned}$$

and

$$\begin{aligned} \text{vec}(dB \odot (Q_{t-1} - \Gamma)) &= \text{vec}((Q_{t-1} - \Gamma) \odot dB) \\ &= \delta[Q_{t-1} - \Gamma] \text{vec}(dB) \\ &= \delta[Q_{t-1} - \Gamma] D_n \text{vech}(dB). \end{aligned}$$

The differential also has a recursive structure, so

$$\text{vec}(B \odot dQ_{t-1}) = \delta[B] \text{vec}(dQ_{t-1}).$$

---

<sup>1</sup> Obviously, in a realistic application the conditional expectation of the observables  $x_t$  has to be modelled somehow.

Finally, we must take into account the dependence of elements  $Q_t$  from the conditional variances.

$$\begin{aligned} \text{vec} (A \odot d (\eta_{t-1} \eta'_{t-1})) &= \delta [A] \text{vec} (d (\eta_{t-1} \eta'_{t-1})) \\ &= -\frac{1}{2} \delta [A] \left[ (\eta_{t-1} \eta'_{t-1}) \otimes V_{t-1}^{-1} \right] G_n dv_{t-1} + \\ &\quad -\frac{1}{2} \delta [A] \left[ V_{t-1}^{-1} \otimes (\eta_{t-1} \eta'_{t-1}) \right] G_n dv_{t-1}. \end{aligned}$$

By an analogous argument as in Equation (18), the above can also be written as

$$\text{vec} (A \odot d (\eta_{t-1} \eta'_{t-1})) = -\delta [A] N_n \left[ V_{t-1}^{-1} \otimes (\eta_{t-1} \eta'_{t-1}) \right] G_n dv_{t-1}.$$

In order to use a more manageable notation, we introduce the following matrices:

$$\begin{aligned} J_1 &= \delta [t_n t'_n - A - B] S_n, & (19) \\ \left( n^2 \times \frac{n(n-1)}{2} \right) \end{aligned}$$

$$\begin{aligned} J_{2,t} &= \delta [\eta_{t-1} \eta'_{t-1} - \Gamma] D_n, & (20) \\ \left( n^2 \times \frac{n(n+1)}{2} \right) \end{aligned}$$

$$\begin{aligned} J_{3,t} &= \delta [Q_{t-1} - \Gamma] D_n, & (21) \\ \left( n^2 \times \frac{n(n+1)}{2} \right) \end{aligned}$$

$$\begin{aligned} J_{4,t} &= -\delta [A] N_n \left[ V_{t-1}^{-1} \otimes (\eta_{t-1} \eta'_{t-1}) \right] G_n, & (22) \\ \left( n^2 \times n \right) \end{aligned}$$

$$\begin{aligned} J_5 &= \delta [B]. & (23) \\ \left( n^2 \times n^2 \right) \end{aligned}$$

After substituting them into the differential we obtain

$$\begin{aligned} \text{vec} (dQ_t) &= J_1 \text{vechl} (d\Gamma) + J_{2,t} \text{vech} (dA) + J_{3,t} \text{vech} (dB) + \\ &\quad + J_{4,t} dv_{t-1} + J_5 \text{vec} (dQ_{t-1}). \end{aligned} \tag{24}$$

#### 4. Gradient of the DCC Model

By differentiating Equation (3), the score for the DCC model is seen to equal

$$\frac{\partial l_t}{\partial \theta'} = C_{1,t} \frac{\partial v_t}{\partial \theta'} + C_{2,t} C_{3,t} \frac{\partial \text{vec} (Q_t)}{\partial \theta'}. \tag{25}$$

Furthermore, the derivatives of  $v_t$  with respect to the various parameter matrices are

$$\begin{aligned} \frac{\partial v_t}{\partial \omega'} &= I_n + \mathcal{B} \frac{\partial v_{t-1}}{\partial \omega'}, \\ \frac{\partial v_t}{\partial \text{vec} (\mathcal{A})'} &= W_{1,t} + \mathcal{B} \frac{\partial v_{t-1}}{\partial \text{vec} (\mathcal{A})'}, \\ \frac{\partial v_t}{\partial \text{vec} (\mathcal{B})'} &= W_{2,t} + \mathcal{B} \frac{\partial v_{t-1}}{\partial \text{vec} (\mathcal{B})'}, \\ \frac{\partial v_t}{\partial \text{vechl} (\Gamma)'} &= 0, \\ \frac{\partial v_t}{\partial \text{vech} (A)'} &= 0, \\ \frac{\partial v_t}{\partial \text{vech} (B)'} &= 0. \end{aligned}$$

In the same fashion, the derivatives of  $Q_t$  are

$$\begin{aligned}
 \frac{\partial \text{vec}(Q_t)}{\partial \omega'} &= J_{4,t} \frac{\partial v_{t-1}}{\partial \omega'} + J_5 \frac{\partial \text{vec}(Q_{t-1})}{\partial \omega'}, \\
 \frac{\partial \text{vec}(Q_t)}{\partial \text{vec}(\mathcal{A})'} &= J_{4,t} \frac{\partial v_{t-1}}{\partial \text{vec}(\mathcal{A})'} + J_5 \frac{\partial \text{vec}(Q_{t-1})}{\partial \text{vec}(\mathcal{A})'}, \\
 \frac{\partial \text{vec}(Q_t)}{\partial \text{vec}(\mathcal{B})'} &= J_{4,t} \frac{\partial v_{t-1}}{\partial \text{vec}(\mathcal{B})'} + J_5 \frac{\partial \text{vec}(Q_{t-1})}{\partial \text{vec}(\mathcal{B})'}, \\
 \frac{\partial \text{vec}(Q_t)}{\partial \text{vechl}(\Gamma)'} &= J_1 + J_5 \frac{\partial \text{vec}(Q_{t-1})}{\partial \text{vechl}(\Gamma)'}, \\
 \frac{\partial \text{vec}(Q_t)}{\partial \text{vech}(A)'} &= J_{2,t} + J_5 \frac{\partial \text{vec}(Q_{t-1})}{\partial \text{vech}(A)'}, \\
 \frac{\partial \text{vec}(Q_t)}{\partial \text{vech}(B)'} &= J_{3,t} + J_5 \frac{\partial \text{vec}(Q_{t-1})}{\partial \text{vech}(B)'}.
 \end{aligned}
 \tag{26}$$

We finally represent the gradient in a more compact form as

$$\frac{\partial l_t}{\partial \theta'} = [C_{1,t} : C_{2,t} : C_{3,t}] \left[ \begin{array}{c} \frac{\partial v_t}{\partial \theta'} \\ \frac{\partial \text{vec}(Q_t)}{\partial \theta'} \end{array} \right],
 \tag{27}$$

where

$$\begin{aligned}
 \left[ \begin{array}{c} \frac{\partial v_t}{\partial \theta'} \\ \frac{\partial \text{vec}(Q_t)}{\partial \theta'} \end{array} \right]_{((n+n^2) \times \dim(\theta))} &= \left[ \begin{array}{c} I_n : W_{1,t} : W_{2,t} : \begin{matrix} 0 \\ (n \times (d_4+d_5+d_6)) \end{matrix} \\ 0 : J_1 : J_{2,t} : J_{3,t} \\ (n^2 \times (d_1+d_2+d_3)) \end{array} \right] + \\
 &+ \left[ \begin{array}{cc} \mathcal{B} & 0 \\ J_{4,t} & J_5 \end{array} \right]_{(n \times n^2)} \left[ \begin{array}{c} \frac{\partial v_{t-1}}{\partial \theta'} \\ \frac{\partial \text{vec}(Q_{t-1})}{\partial \theta'} \end{array} \right].
 \end{aligned}
 \tag{28}$$

#### 4.1. Gradient of Other DCC Specifications

Using the previous results, we could derive the gradient of several variants of the DCC model of Engle (2002). In fact, alternative models generally differ from that of Engle (2002), only in the equation governing the correlation dynamic. We provide here an example for the GDCC model of Cappiello et al. (2006), while in the next section we focus on the more recent cDCC model of Aielli (2013).

The differential of the GDCC equation is

$$\begin{aligned}
 dQ_t &= d\Gamma - dA\Gamma A' - Ad\Gamma A' - A\Gamma dA' + \\
 &- dB\Gamma B' - Bd\Gamma B' - B\Gamma dB' + \\
 &+ dA\eta_{t-1}\eta'_{t-1}A' + Ad(\eta_{t-1}\eta'_{t-1})A' + A\eta_{t-1}\eta'_{t-1}dA' + \\
 &+ dBQ_{t-1}B' + BdQ_{t-1}B' + BQ_{t-1}dB'.
 \end{aligned}$$

and taking the vec, with some manipulations and using previous results for the differential of  $d(\eta_{t-1}\eta'_{t-1})$ , we obtain

$$\begin{aligned}
 \text{vec}(dQ_t) &= J_1 \text{vechl}(d\Gamma) + J_{2,t} \text{vech}(dA) + J_{3,t} \text{vech}(dB) + \\
 &+ J_{4,t} dv_{t-1} + J_5 \text{vec}(dQ_{t-1}),
 \end{aligned}
 \tag{29}$$

where

$$J_1 = (I_{n^2} + A \otimes A + B \otimes B) S_n, \tag{30}$$

$$J_{2,t} = 2N_n [(A\Gamma \otimes I_n) + (A\eta_{t-1}\eta'_{t-1} \otimes I_n)], \tag{31}$$

$$J_{3,t} = 2N_n [(B\Gamma \otimes I_n) + (BQ_{t-1} \otimes I_n)], \tag{32}$$

$$J_{4,t} = -\frac{1}{2} (A \otimes A) N_n [V_{t-1}^{-1} \otimes (\eta_{t-1}\eta'_{t-1})] G_n, \tag{33}$$

$$J_5 = B \otimes B. \tag{34}$$

Finally, the gradient has exactly the same form as in (27) with the  $J$  matrices defined in (30)–(34).

#### 4.2. Gradient of the cDCC Model

The cDCC model of Aielli (2013) differs from the DCC model of Engle (2002) just in the dynamic equation of the  $Q_t$  matrix. Thus, the two gradients are very similar: the differential of  $Q_t$  is equal to

$$\begin{aligned} dQ_t = & (\iota_n \iota_n' - A - B) \odot d\Gamma + dA \odot (\tilde{Q}_{t-1}^{1/2} \eta_{t-1} \eta'_{t-1} \tilde{Q}_{t-1}^{1/2} - \Gamma) + \\ & + dB \odot (Q_{t-1} - \Gamma) + B \odot dQ_{t-1} + A \odot d(\tilde{Q}_{t-1}^{1/2} \eta_{t-1} \eta'_{t-1} \tilde{Q}_{t-1}^{1/2}). \end{aligned}$$

The difference from the original DCC by Engle (2002) therefore comes from two terms:  $dA \odot (\tilde{Q}_{t-1}^{1/2} \eta_{t-1} \eta'_{t-1} \tilde{Q}_{t-1}^{1/2} - \Gamma)$  and  $d(\tilde{Q}_{t-1}^{1/2} \eta_{t-1} \eta'_{t-1} \tilde{Q}_{t-1}^{1/2})$ . For the former, we simply have

$$\tilde{J}_{2,t} = \delta \left[ \tilde{Q}_{t-1}^{1/2} \eta_{t-1} \eta'_{t-1} \tilde{Q}_{t-1}^{1/2} - \Gamma \right] D_n. \tag{35}$$

$(n^2 \times \frac{n(n+1)}{2})$

On the contrary, the differential for the dynamic correlations shocks must be entirely reconsidered. First note that

$$\begin{aligned} d(\tilde{Q}_{t-1}^{1/2} \eta_{t-1} \eta'_{t-1} \tilde{Q}_{t-1}^{1/2}) = & d\tilde{Q}_{t-1}^{1/2} \eta_{t-1} \eta'_{t-1} \tilde{Q}_{t-1}^{1/2} + \\ & + \tilde{Q}_{t-1}^{1/2} d(\eta_{t-1} \eta'_{t-1}) \tilde{Q}_{t-1}^{1/2} + \\ & + \tilde{Q}_{t-1}^{1/2} \eta_{t-1} \eta'_{t-1} d\tilde{Q}_{t-1}^{1/2}. \end{aligned}$$

Taking the vec and using the operator’s properties we obtain

$$\begin{aligned} \text{vec} \left( d(\tilde{Q}_{t-1}^{1/2} \eta_{t-1} \eta'_{t-1} \tilde{Q}_{t-1}^{1/2}) \right) = & (\eta_{t-1} \eta'_{t-1} \tilde{Q}_{t-1}^{1/2} \otimes I_n) \text{vec} \left( d\tilde{Q}_{t-1}^{1/2} \right) + \\ & + (\tilde{Q}_{t-1}^{1/2} \otimes \tilde{Q}_{t-1}^{1/2}) \text{vec} \left( d(\eta_{t-1} \eta'_{t-1}) \right) + \\ & + (I_n \otimes \tilde{Q}_{t-1}^{1/2} \eta_{t-1} \eta'_{t-1}) \text{vec} \left( d\tilde{Q}_{t-1}^{1/2} \right). \end{aligned}$$

Using previous results we also have

$$\begin{aligned} \text{vec} \left( d(\tilde{Q}_{t-1}^{1/2} \eta_{t-1} \eta'_{t-1} \tilde{Q}_{t-1}^{1/2}) \right) = & (\tilde{Q}_{t-1}^{1/2} \otimes \tilde{Q}_{t-1}^{1/2}) \text{vec} \left( d(\eta_{t-1} \eta'_{t-1}) \right) + \\ & + \frac{1}{2} \left[ (\eta_{t-1} \eta'_{t-1} \tilde{Q}_{t-1}^{1/2} \otimes I_n) + (I_n \otimes \tilde{Q}_{t-1}^{1/2} \eta_{t-1} \eta'_{t-1}) \right] \tilde{Q}_{t-1}^{-1/2} \text{vec} \left( dQ_{t-1} \right). \end{aligned}$$

Using the last equation we can reconsider the recursive element of the differential

$$\begin{aligned} \text{vec} \left( A \odot d \left( \tilde{Q}_{t-1}^{1/2} \eta_{t-1}' \eta_{t-1}' \tilde{Q}_{t-1}^{1/2} \right) \right) &= \delta [A] \text{vec} \left( d \left( \tilde{Q}_{t-1}^{1/2} \eta_{t-1}' \eta_{t-1}' \tilde{Q}_{t-1}^{1/2} \right) \right) = \\ &= -\frac{1}{2} \delta [A] \left( \tilde{Q}_{t-1}^{1/2} \otimes \tilde{Q}_{t-1}^{1/2} \right) \left[ \left( \eta_{t-1}' \eta_{t-1}' \right) \otimes V_{t-1}^{-1} \right] G_n dv_{t-1} + \\ &\quad -\frac{1}{2} \delta [A] \left( \tilde{Q}_{t-1}^{1/2} \otimes \tilde{Q}_{t-1}^{1/2} \right) \left[ V_{t-1}^{-1} \otimes \left( \eta_{t-1}' \eta_{t-1}' \right) \right] G_n dv_{t-1} + \\ &+ \frac{1}{2} \delta [A] \left[ \left( \eta_{t-1}' \eta_{t-1}' \tilde{Q}_{t-1}^{1/2} \otimes I_n \right) + \left( I_n \otimes \tilde{Q}_{t-1}^{1/2} \eta_{t-1}' \eta_{t-1}' \right) \right] \tilde{Q}_{t-1}^{-1/2} \text{vec} (dQ_{t-1}). \end{aligned}$$

Taking into account these additional elements, and algebraic equalities adopted in the previous section, we have:

$$\begin{aligned} \tilde{J}_{4,t} &= -\delta [A] \left( \tilde{Q}_{t-1}^{1/2} \otimes \tilde{Q}_{t-1}^{1/2} \right) N_n \left[ V_{t-1}^{-1} \otimes \left( \eta_{t-1}' \eta_{t-1}' \right) \right] G_n, \\ (n^2 \times n) & \\ \tilde{J}_{5,t} &= \delta [B] + \delta [A] \times N_n \left( \eta_{t-1}' \eta_{t-1}' \tilde{Q}_{t-1}^{1/2} \otimes I_n \right) \tilde{Q}_{t-1}^{-1/2}, \\ (n^2 \times n^2) & \end{aligned} \tag{36}$$

and we stress that the last matrix is now time-varying.

Finally, the cDCC gradient has the following compact form (which is identical to that of the DCC model apart for the content of the four matrices with a tilde)

$$\begin{aligned} \frac{\partial l_t}{\partial \theta'} &= [C_{1,t} : C_{2,t} C_{3,t}] \left[ \begin{array}{c} \frac{\partial v_t}{\partial \theta'} \\ \frac{\partial \text{vec}(Q_t)}{\partial \theta'} \end{array} \right] \\ & \\ \left[ \begin{array}{c} \frac{\partial v_t}{\partial \theta'} \\ \frac{\partial \text{vec}(Q_t)}{\partial \theta'} \end{array} \right] &= \left[ \begin{array}{c} I_n : W_{1,t} : W_{2,t} : 0 \\ (n \times (d_4 + d_5 + d_6)) \\ 0 : J_1 : \tilde{J}_{2,t} : J_{3,t} \\ (n^2 \times (d_1 + d_2 + d_3)) \end{array} \right] + \\ & \\ &+ \left[ \begin{array}{cc} \mathcal{B} & 0 \\ \tilde{J}_{4,t} & \tilde{J}_{5,t} \end{array} \right] \left[ \begin{array}{c} \frac{\partial v_{t-1}}{\partial \theta'} \\ \frac{\partial \text{vec}(Q_{t-1})}{\partial \theta'} \end{array} \right]. \end{aligned} \tag{37}$$

We note that the differences between the gradient of the DCC of Engle (2002) and of the cDCC of Aielli (2013) are confined to the matrices  $\tilde{J}_{4,t}$ ,  $\tilde{J}_{5,t}$ , and  $\tilde{J}_{6,t}$ , where the last one is dynamic for the cDCC while it is static for the DCC. As a result, the implementation of the analytical gradient for the cDCC model would be computationally (marginally) more complex than that for the DCC model.

### 5. Two-Step Estimation of Dcc-Like Models

The previous sections provide the analytical expressions for the gradient of selected DCC specifications. The availability of analytical gradients has relevant advantages from the model estimation point of view. In fact, they might bring about substantial savings in the CPU time needed for model estimation, particularly so when the model dimension  $n$  is large. However, some qualifications have to be made.

When estimating DCC-type models, most practitioners refrain from maximising the full log-likelihood and rely instead on a two-step approach put forward by Engle (2002). In the first step, a battery of  $n$  univariate GARCH models is estimated, one for each series; the second step involves the maximisation of the log-likelihood (3) over the parameter matrices  $\Gamma$ ,  $A$  and  $B$  by taking  $\varepsilon_t$  and  $v_t$  as given and using the estimated residuals and conditional variances in their stead.

Given the structure of the gradient, the two-step approach has adverse consequences on efficiency. In fact, estimating the conditional variance parameters separately from the conditional correlation parameters leads to using numerical procedures for the evaluation of the gradient and the Hessian that neglect the dependence of the conditional variance parameters' derivatives on the conditional

correlation. The availability of the DCC model gradient might be of help even in the case we plan to maintain a two-step estimation approach. In fact, the Maximisation by Parts (MbP) approach of [Fan et al. \(2015\)](#), introduces an iterative approach for a general case that includes DCC-type models. The framework of MbP requires that a log-likelihood function depends on two sets of parameters and can be decomposed as

$$L_T(\theta_1, \theta_2) = L_{1,T}(\theta_1) + L_{2,T}(\theta_1, \theta_2). \tag{38}$$

Imposing the first order conditions leads to

$$\frac{\partial L_T(\theta_1, \theta_2)}{\partial \theta'_1} = \frac{\partial L_{1,T}(\theta_1)}{\partial \theta'_1} + \frac{\partial L_{2,T}(\theta_1, \theta_2)}{\partial \theta'_1} = 0, \tag{39}$$

$$\frac{\partial L_T(\theta_1, \theta_2)}{\partial \theta'_2} = \frac{\partial L_{2,T}(\theta_1, \theta_2)}{\partial \theta'_2} = 0. \tag{40}$$

[Fan et al. \(2015\)](#) suggest estimating the parameters according to an iterative scheme that takes advantage of the fact that one of the two parameters set appears only in one of the two components of the likelihood, and tries at the same time to avoid neglecting the second term of  $\frac{\partial L_T(\theta_1, \theta_2)}{\partial \theta'_1}$ . The MbP approach estimates parameters iterating between the two following systems of first order conditions, where  $j$  is the iteration index:

$$\frac{\partial L_{1,T}(\theta_1)}{\partial \theta'_1} + \frac{\partial L_{2,T}(\theta_1, \hat{\theta}_{j-1,2})}{\partial \theta'_1} = 0, \tag{41}$$

$$\frac{\partial L_{2,T}(\hat{\theta}_{j,1}, \theta_2)}{\partial \theta'_2} = 0. \tag{42}$$

The DCC log-likelihood is coherent with that framework, as [Engle \(2002\)](#) shows that we can separate the conditional variance contribution to the overall likelihood from the conditional correlation contribution. Moreover, if we let the conditional correlation parameters to be included in the set  $\theta_1$  and the conditional variance parameters to be included in  $\theta_2$ , coherently with our previous derivations, we have that the conditional variance parameters enter in the log-likelihood of the conditional correlations. The availability of the analytical gradient for a DCC model allows adopting the MbP approach with a computational advantage compared to the use of numerical methods.

However, as the gradient with respect to the conditional variance parameters depend on the GARCH-type specifications adopted, we concentrate from now on only on the implementation of the gradient with respect to the conditional correlation parameters.

## 6. Software Implementation of Analytical Gradients

The implementation of computer code to translate the formulae of the previous section into actual algorithms is not trivial. First, it should be noted that for efficient numerical implementation of the score several special matrices used here (such as  $D_n$ ,  $K_{mm}$  or  $\Delta_n$ ) are selection/permutation matrices. Therefore, pre- or post-multiplication of a matrix (say,  $A$ ) by a special matrix simply amounts to selecting/reordering the rows and columns of  $A$ , thereby avoiding floating-point operations.

Moreover, the full generality of the law of motion for  $Q_t$  as provided in Equation (1) is rarely needed, and practitioners have focused on several sub-cases, which allow for different reparametrisations, some of which involve parameter restrictions. In all these cases, the parameter vector  $\theta$  is re-expressed as  $\theta = \theta(\psi)$ , where  $\psi$  contains the unrestricted parameters for the numerical maximisation algorithm. We present here the analysis for the DCC model by [Engle \(2002\)](#) but similar adaptations are possible for the other DCC specifications as well.

The first example of these restrictions involves the  $\Gamma$  matrix in (1), which could generally be parametrised as a positive semidefinite symmetric matrix with ones on the diagonal. However, the  $\frac{n(n-1)}{2}$  free parameters that  $\Gamma$  contains are often constrained in such a way that  $\Gamma$  equals some

prescribed matrix, usually obtained from the data. This idea, introduced in Engle and Mezrich (1996), is very common among practitioners and goes under the name of “variance targeting”.<sup>2</sup>

Moreover, the  $A$  and  $B$  matrices are also often reparametrised as well in several ways:

**Cholesky factorisation** The  $A$  and  $B$  matrices are expressed as

$$A = H_A H_A', \quad B = H_B H_B', \tag{43}$$

where  $H_A$  and  $H_B$  are lower triangular matrices, ensuring that  $A$  and  $B$  are positive semi-definite. In this case,  $\psi' = [\text{vech}(H_A)' : \text{vech}(H_B)']$  and  $\theta = \theta(\psi)$  is a bijective transformation and no constraints are set.

**Rank-1 specification** The  $A$  and  $B$  matrices are expressed as

$$A = \mathbf{a}\mathbf{a}', \quad B = \mathbf{b}\mathbf{b}', \tag{44}$$

where  $\mathbf{a}$  and  $\mathbf{b}$  are column  $n$ -vectors. Note that this is a special case of the one above, where  $A$  and  $B$  are restricted to be rank 1 matrices. Here,  $\psi' = [\mathbf{a}' : \mathbf{b}']$  is a  $2n \times 1$  vector and the transformation is not invertible. This parametrisation implies  $n^2 - n$  constraints; as a consequence, the dimension of  $\psi$  is  $O(n)$ .

**Scalar DCC** The  $A$  and  $B$  matrices are expressed as

$$A = \alpha \cdot \mathbf{1}_n \mathbf{1}_n', \quad B = \beta \cdot \mathbf{1}_n \mathbf{1}_n', \tag{45}$$

where  $\alpha$  and  $\beta$  are positive scalars. Note that this is a restricted case of the above, where  $\mathbf{a} = \sqrt{\alpha} \cdot \mathbf{1}_n$  and  $\mathbf{b} = \sqrt{\beta} \cdot \mathbf{1}_n$ . Here,  $\psi' = [\alpha : \beta]$  (a  $2 \times 1$  vector) and  $n^2 + n - 2$  constraints are implied; hence, the dimension of  $\psi$  is fixed at 2, irrespective of  $n$ .

All these reparametrisations have the advantage that the unknown element of  $\psi$  can vary freely and no numerical check on the symmetry and positive-semidefiniteness on  $A$  and  $B$  is required.

The analytical gradient with respect to the unconstrained parameters  $\psi$  is readily obtained by post-multiplying Equation (4) by the Jacobian term  $\Psi = \frac{\partial \theta}{\partial \psi}$ , that could be written more extensively as

$$\Psi_{\dim(\theta) \times \dim(\psi)} = \frac{\partial \theta}{\partial \psi} = \begin{bmatrix} \frac{\partial \text{vech} \Gamma}{\partial \psi} \\ \frac{\partial \text{vech} A}{\partial \psi} \\ \frac{\partial \text{vech} B}{\partial \psi} \end{bmatrix} = \begin{bmatrix} \Psi_\Gamma \\ \Psi_A \\ \Psi_B \end{bmatrix}. \tag{46}$$

Note that the matrix  $\Psi$  is time-invariant, so it can be computed just once: in practice, once a loop over  $t = 1 \dots T$  has been run on Equation (25) and the results have been stored into a  $T \times \dim(\theta)$  matrix, all is needed for computing the analytical score with respect to  $\psi$  is post-multiplying this matrix by  $\Psi$ .

If the  $\Gamma$  matrix is estimated without restrictions, then  $\Psi_\Gamma = I_{\frac{n(n-1)}{2}}$ ; instead, under variance targeting  $\Gamma$  contains no parameters at all and  $\Psi_\Gamma$  simply vanishes.

As for  $\Psi_A$ , for the three cases listed above, we have that,

$$\text{dvech}(A) = \begin{cases} 2L_n [N_n (H_A \otimes I_n) D_n] \text{dvech}(H_A) & \text{Cholesky} \\ 2L_n [N_n (\mathbf{a} \otimes I_n)] d\mathbf{a} & \text{rank-1} \\ L_n \mathbf{1}_n^2 d\alpha = \mathbf{1}_{\frac{n(n+1)}{2}} d\alpha & \text{scalar} \end{cases}$$

<sup>2</sup> A possible alternative reparametrisations, based on hyperbolic trigonometric functions, is the one proposed in Pelletier (2006).

so that

$$\Psi_A = \begin{cases} 2L_n [N_n (H_A \otimes I_n) D_n] & \text{Cholesky} \\ 2L_n [N_n (\mathbf{a} \otimes I_n)] & \text{rank-1} \\ \boldsymbol{\iota}_{\frac{n(n+1)}{2}} & \text{scalar} \end{cases} \quad (47)$$

and similar expressions hold, *mutatis mutandis*, for the matrix  $B$ .

### 7. Empirical Example

In order to give the reader an idea of the computational advantages that analytical scores bring about, we ran the following experiment: we generated random data for  $T = 512, 1024, 2048$  and evaluated the score at a given  $\psi$ , for four different DCC specification.

We evaluate CPU time and precision in the computation of the derivative of  $l(\theta; x_t)$ ; this task is accomplished by taking a specific data point  $x_t, t = 1 \dots T$ ; note that the evaluation of CPU time and precision has nothing to do with the way the specific sample  $x_t$  was generated, as long as the point  $x_t$  belongs to the support of the density whose log-likelihood we are evaluating.

However, to make sure this is the case, we ran the experiment several times, on data points that belong to the support of densities that can conceivably be of interest to the practitioners, such as such as those that represent persistency at high frequencies, jumps and long-memory (see, for example, [Gradojevic et al. 2020](#)),<sup>3</sup> we found no appreciable differences and the additional tables, not included in the paper for the sake of completeness but available upon request from the authors, are completely in line with the one reported in this section.

The DGPs we chose can be ordered in increasing order of restrictedness; we use the following sets of parameters:

1. Unrestricted (no transformation):  $\Gamma = I_n, A = 0.1 \cdot P_n$  and  $B = 0.9 \cdot P_n$ , where  $P_n = I_n + n^{-1} \boldsymbol{\iota}_n \boldsymbol{\iota}_n'$ ; therefore,

$$\psi = \left[ \mathbf{0}_{\frac{n(n-1)}{2}} : 0.1 \cdot \text{vech}(P_n) : 0.9 \cdot \text{vech}(P_n) \right]'$$

2. Cholesky:  $\Gamma = I_n, H_A = 0.1 \cdot H_n$  and  $H_B = 0.9 \cdot H_n$ , where  $H_n$  is a lower-triangular matrix equal to  $P_n$  in its non-zero elements; therefore,

$$\psi = \left[ \mathbf{0}_{\frac{n(n-1)}{2}} : 0.1 \cdot \text{vech}(P_n) : 0.9 \cdot \text{vech}(P_n) \right]'$$

3. rank-1:  $\Gamma = I_n, \mathbf{a} = 0.1 \cdot \boldsymbol{\iota}_n$  and  $\mathbf{b} = 0.9 \cdot \boldsymbol{\iota}_n$ ; therefore,

$$\psi = \left[ \mathbf{0}_{\frac{n(n-1)}{2}} : 0.1 \cdot \boldsymbol{\iota}_n : 0.9 \cdot \boldsymbol{\iota}_n \right]'$$

4. scalar:  $\Gamma = I_n, \alpha = 0.1$  and  $\beta = 0.9$ ; therefore,

$$\psi = \left[ \mathbf{0}_{\frac{n(n-1)}{2}} : 0.1 : 0.9 \right]'$$

The same specifications were also analysed under the variance targeting setup, where of course the first block of  $\psi$  drops out.

The score was computed both analytically and numerically. For each parameter configuration, Tables 1 and 2 report two indicators, respectively:

---

<sup>3</sup> We thank an anonymous referee for bringing up this point.



1. the CPU timings ratio  $\rho = \frac{t_N}{t_A}$ , where  $t_N$  and  $t_A$  are the CPU times for the numerical and analytical gradient, respectively;
2. an indicator of the numerical accuracy of numerical derivatives,  $d^{\max}$ , given the max absolute difference between numerical and analytical gradients.<sup>4</sup>

Results show quite clearly that analytical differentiation is uniformly advantageous in terms of precision, and increasingly so as  $\dim(\psi)$  gets large, either because of larger  $n$  or fewer constraints on the model. The quality of numerical approximation also degrades uniformly as the sample size  $T$  grows. The comparison was performed by using the central difference formula (see for example, [Epperson 2013](#), Section 2.2) for numerical derivatives; obviously, more sophisticated methods, such as Richardson extrapolation, would yield more accuracy but would also imply heavier CPU usage.

On the contrary, analytical derivatives seem to yield a speed premium only when  $\psi$  is  $O(n^2)$  or more, as the formulae presented in this paper entail a computational overhead that seems roughly to be  $O(n)$ . As can be seen, analytical derivatives dominate numerical ones in all those situations when the number of unrestricted parameters gets large. Interestingly, the opposite happens when constraints are pushed to the extreme: in the scalar model with variance targeting, the number of unrestricted parameters is fixed at 2 no matter what the model size is, and analytical derivatives are actually detrimental, and increasingly so as  $n$  gets larger. Sample size does not seem to make much difference, instead.

For any realistic estimation problem, therefore, the combined effect of increased precision and quicker computation make it advisable to employ analytical derivatives any time the model being estimated is more general than the simple scalar model: even if CPU time were roughly equivalent in the two cases, the deterioration in precision one gets with numerical derivatives would imply a substantial increase in the number of iterations needed to reach the maximum of the log-likelihood function.

A word of warning: our results should be considered only suggestive, since the actual hardware and software setup may make quite a large difference: for example, the results reported in [Table 2](#) were obtained on a Linux machine with 20 physical CPU cores, using the `gretl` package linking to the OpenBLAS matrix library ([Wang et al. 2013](#)). Results may be different on other hardware/software combinations.

---

<sup>4</sup> One may think that a discrepancy should not mean in itself that numerical derivatives are “wrong” and analytical ones are “right”. The analytical results were carefully checked for small values of  $n$  and were found to be correct; since the formulae employed in the score calculations in [Sections 3 and 4](#) do not contain operations that should be particularly sensitive to rounding errors, we are confident in their validity. Nevertheless, to be on the safe side, we repeated the comparison using a slower but more accurate algorithm for computing numerical derivatives and found that  $d^{\max}$  decreased uniformly, as expected.

Table 1. Maximum distance indices ( $d^{\max}$ ).

Model	1: No Transformation						2: Cholesky					
	No Targeting			Targeting			No Targeting			Targeting		
T	512	1024	2048	512	1024	2048	512	1024	2048	512	1024	2048
n = 2	0.0003	0.0003	0.0025	0.0003	0.0003	0.0032	0.0002	0.0002	0.0020	0.0002	0.0002	0.0020
n = 3	0.0012	0.0025	0.0052	0.0012	0.0025	0.0078	0.0010	0.0029	0.0035	0.0010	0.0029	0.0035
n = 4	0.0014	0.0052	0.0082	0.0014	0.0052	0.0164	0.0017	0.0051	0.0123	0.0017	0.0051	0.0123
n = 5	0.0040	0.0077	0.0193	0.0040	0.0077	0.0391	0.0049	0.0086	0.0304	0.0049	0.0086	0.0304
n = 6	0.0044	0.0239	0.0410	0.0044	0.0239	0.0564	0.0053	0.0124	0.0648	0.0053	0.0124	0.0648
n = 7	0.0048	0.0218	0.0709	0.0048	0.0218	0.0580	0.0075	0.0265	0.0622	0.0075	0.0265	0.0622
n = 8	0.0126	0.0377	0.1104	0.0126	0.0377	0.0776	0.0104	0.0286	0.0925	0.0104	0.0286	0.0925
n = 9	0.0129	0.0570	0.1127	0.0129	0.0570	0.0927	0.0131	0.0375	0.1216	0.0131	0.0375	0.1216
n = 10	0.0253	0.0654	0.1746	0.0253	0.0654	0.1492	0.0212	0.0621	0.2129	0.0212	0.0621	0.2129
n = 15	0.0399	0.1289	0.4150	0.0399	0.1289	0.3692	0.0395	0.1086	0.3534	0.0395	0.1086	0.3534
n = 20	0.0927	0.2619	0.8656	0.0927	0.2619	0.7093	0.0919	0.2900	0.6051	0.0919	0.2900	0.6052
n = 25	0.1850	0.4536	1.1483	0.1850	0.4536	0.9630	0.1321	0.4474	0.9838	0.1321	0.4474	0.9838

model	3: Rank-1						4: Scalar					
	No Targeting			Targeting			No Targeting			Targeting		
T	512	1024	2048	512	1024	2048	512	1024	2048	512	1024	2048
n = 2	$8.3 \times 10^{-5}$	0.0002	0.0007	$8.3 \times 10^{-5}$	0.0002	0.0007	0.0002	0.0002	0.0006	0.0002	0.0002	0.0006
n = 3	$6.6 \times 10^{-5}$	0.0005	0.0014	$6.6 \times 10^{-5}$	0.0005	0.0014	$1.9 \times 10^{-5}$	0.0004	0.0006	$1.9 \times 10^{-5}$	0.0004	0.0006
n = 4	0.0002	0.0009	0.0008	0.0002	0.0009	0.0008	$3.5 \times 10^{-5}$	0.0004	0.0031	$3.5 \times 10^{-5}$	0.0004	0.0031
n = 5	0.0004	0.0009	0.0022	0.0004	0.0009	0.0022	$1.4 \times 10^{-5}$	0.0005	$5.4 \times 10^{-5}$	$1.4 \times 10^{-5}$	0.0005	$5.4 \times 10^{-5}$
n = 6	0.0002	0.0012	0.0045	0.0002	0.0012	0.0045	0.0003	0.0005	0.0003	0.0003	0.0005	0.0003
n = 7	0.0006	0.0018	0.0026	0.0006	0.0018	0.0026	0.0007	0.0017	0.0026	0.0007	0.0017	0.0026
n = 8	0.0004	0.0016	0.0043	0.0004	0.0016	0.0043	0.0010	0.0014	0.0006	0.0010	0.0014	0.0006
n = 9	0.0009	0.0026	0.0058	0.0009	0.0026	0.0058	$3.3 \times 10^{-5}$	0.0005	0.0018	$3.3 \times 10^{-5}$	0.0005	0.0018
n = 10	0.0006	0.0022	0.0031	0.0006	0.0022	0.0031	0.0005	0.0007	0.0055	0.0005	0.0007	0.0055
n = 15	0.0010	0.0039	0.0152	0.0010	0.0039	0.0152	0.0005	0.0020	0.0005	0.0005	0.0020	0.0005
n = 20	0.0018	0.0063	0.0112	0.0018	0.0063	0.0112	0.0026	0.0031	0.0311	0.0026	0.0031	0.0311
n = 25	0.0022	0.0055	0.0174	0.0022	0.0055	0.0174	0.0065	0.0329	0.0254	0.0065	0.0329	0.0254

T is sample size, while n is the number of series in the model.

Table 2. Time ratios  $t_N/t_A$ .

Model	1: No Transformation						2: Cholesky					
	No Targeting			Targeting			No Targeting			Targeting		
T	512	1024	2048	512	1024	2048	512	1024	2048	512	1024	2048
n = 2	1.47	1.53	1.48	1.47	1.42	1.27	1.59	1.49	6.54	1.51	1.43	1.40
n = 3	2.96	3.03	2.94	2.77	2.67	2.49	3.12	2.96	2.98	2.83	2.73	2.61
n = 4	5.03	5.07	4.89	4.42	4.44	3.92	5.21	4.96	5.03	4.57	4.33	4.28
n = 5	7.55	7.59	7.30	6.41	6.44	5.97	7.60	7.36	7.49	6.60	6.30	6.17
n = 6	5.57	5.68	5.55	4.71	4.73	4.25	5.58	5.54	5.75	5.14	4.71	4.53
n = 7	7.31	6.91	7.01	5.88	5.91	5.40	7.04	7.06	7.12	5.94	5.84	5.50
n = 8	8.54	8.24	8.33	6.91	7.46	6.33	8.45	8.48	8.61	7.06	6.90	6.45
n = 9	9.47	9.23	9.41	7.73	8.37	6.60	9.51	9.65	9.63	7.85	7.74	7.02
n = 10	10.41	9.79	9.47	8.06	8.26	8.14	10.72	9.96	10.40	8.19	8.11	7.69
n = 15	11.65	11.14	12.42	8.78	9.03	8.33	12.25	11.48	11.57	9.14	8.97	8.33
n = 20	8.84	8.45	8.89	6.13	6.03	6.25	9.02	8.71	8.93	6.40	5.97	6.38
n = 25	7.50	7.71	7.68	5.06	4.99	5.13	7.72	7.63	7.54	5.88	5.19	5.13

model	3: Rank-1						4: Scalar					
	No Targeting			Targeting			No Targeting			Targeting		
T	512	1024	2048	512	1024	2048	512	1024	2048	512	1024	2048
n = 2	1.11	1.07	1.09	0.93	1.01	0.92	0.70	0.68	0.71	0.58	0.55	0.56
n = 3	1.87	1.90	1.80	1.30	1.32	1.42	1.06	1.04	1.07	0.54	0.53	0.55
n = 4	2.78	3.15	1.32	1.71	1.59	1.83	1.68	1.57	1.67	0.55	0.54	0.54
n = 5	3.94	4.33	3.78	2.07	1.97	2.19	2.37	2.26	2.32	0.53	0.51	0.53
n = 6	2.73	2.61	2.69	1.33	1.26	1.40	1.71	1.67	1.80	0.27	0.28	0.28
n = 7	3.35	3.14	3.31	1.41	1.35	1.52	2.15	2.10	2.16	0.25	0.26	0.26
n = 8	3.82	4.05	3.83	1.53	1.45	1.58	2.60	2.54	2.57	0.24	0.24	0.24
n = 9	4.14	3.97	4.12	1.50	1.30	1.61	2.89	2.82	2.90	0.21	0.21	0.22
n = 10	4.48	4.19	4.42	1.51	1.39	1.50	3.06	2.93	3.14	0.19	0.18	0.19
n = 15	4.48	4.36	4.76	1.07	1.02	1.15	3.44	3.55	2.02	0.09	0.09	0.10
n = 20	3.33	3.41	3.35	0.60	0.63	0.58	2.79	2.69	3.13	0.04	0.04	0.04
n = 25	2.87	2.98	2.72	0.42	0.42	0.40	2.45	2.44	2.48	0.02	0.02	0.02

T is sample size, while n is the number of series in the model.

### 8. Concluding Remarks

The formulae we developed make it possible to compute the analytical score for the DCC model quickly and efficiently. Compared to the numerical derivatives modern statistical packages provide, the advantages are twofold: increased precision and computational speed.

However, while increased precision depends rather uniformly on the size of the model being estimated, the effects on CPU time are worth some more detailed comment: generally speaking, analytical derivatives are especially beneficial for larger models when the parameters of the law of motion for the conditional correlation matrix are not overly constrained. In the extreme case of the scalar DCC model with covariance targeting, the computational overhead implied by analytically differentiating the log-likelihood offsets completely the advantages. In all other cases, analytical derivatives provide substantial savings in computing time, often dramatic. Sample size, on the other hand, does not seem to affect the relative speed of analytical versus numerical gradient calculation very much.

In the light of the enormous importance of modelling covariance for large numbers of variables using parametric structures which do not impose an excessive number of constraints, the usage of analytical derivatives seems to be imperative in all cases when modelling accurately the time profile of conditional correlations is essential.

Such situations arise frequently in practice, and a few examples can be given: first, conditional correlation models are relevant for risk management, particularly so when the portfolio includes a great number of assets. Second, these models may prove highly beneficial in asset allocation problems,

since predictions on both the conditional mean and the conditional covariance matrix are required to determine the optimal portfolio composition; in these cases the number of assets in the investment universe is often rather large, so computational complexity is likely to be a very real issue. Third, the possibility of estimating full DCC models in medium-sized problems makes it possible to account for spillovers and interdependence between conditional variances and within conditional correlations, and thereby analyze the diffusion of shocks within and/or between financial markets in greater detail.

**Author Contributions:** Conceptualization, M.C.; methodology, M.C. and R.J.L.; software, R.J.L. and G.P.; investigation, G.P.; data curation, R.J.L. and G.P.; writing—original draft preparation, M.C.; writing—review and editing, M.C., R.J.L. and G.P. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

- Aielli, Gian Piero. 2013. Dynamic conditional correlation: On properties and estimation. *Journal of Business & Economic Statistics* 31: 282–99.
- Bauwens, Luc, Sebastien Laurent, and Jeroen K. Rombouts. 2006. Multivariate GARCH models: A survey. *Journal of Applied Econometrics* 21: 79–109. [\[CrossRef\]](#)
- Bollerslev, Tim. 1990. Modelling the coherence in short-run nominal exchange rates: A multivariate generalized ARCH model. *Review of Economics and Statistics* 72: 498–505. [\[CrossRef\]](#)
- Caporin, Massimiliano and Michael McAleer. 2008. Scalar BEKK and Indirect DCC. *Journal of Forecasting* 27: 537–49. [\[CrossRef\]](#)
- Caporin, Massimiliano and Paolo Paruolo. 2015. Proximity-structured multivariate volatility models. *Econometric Reviews* 34: 559–93. [\[CrossRef\]](#)
- Cappiello, Lorenzo, Robert F Engle, and Kevin Sheppard. 2006. Asymmetric dynamics in the correlations of global equity and bond returns. *Journal of Financial Econometrics* 4: 537–72. [\[CrossRef\]](#)
- Engle, Robert. 2002. Dynamic conditional correlation: A simple class of multivariate generalized autoregressive conditional heteroskedasticity models. *Journal of Business & Economic Statistics* 20: 339–50.
- Engle, Robert F, and Kenneth F Kroner. 1995. Multivariate simultaneous generalized ARCH. *Econometric Theory* 11: 122–50. [\[CrossRef\]](#)
- Engle, Robert F, and J. Mezrich. 1996. GARCH for Groups. *Risk* 9: 36–40.
- Engle, Robert F, and Kevin Sheppard. 2001. *Theoretical and Empirical Properties of Dynamic Conditional Correlation Multivariate GARCH*. Technical Report. Cambridge: National Bureau of Economic Research.
- Epperson, James F. 2013. *An Introduction to Numerical Methods and Analysis*, 2nd ed. Hoboken: Wiley Publishing.
- Fan, Yanqin, Sergio Pastorello, and Eric Renault. 2015. Maximization by parts in extremum estimation. *The Econometrics Journal* 18: 147–71. [\[CrossRef\]](#)
- Gradojevic, Nikola, Deniz Erdemlioglu, and Ramazan Gençay. 2020. A new wavelet-based ultra-high-frequency analysis of triangular currency arbitrage. *Economic Modelling* 85: 57–73. [\[CrossRef\]](#)
- Hafner, Christian M, and Helmut Herwartz. 2008. Analytical quasi maximum likelihood inference in multivariate volatility models. *Metrika* 67: 219–39. [\[CrossRef\]](#)
- Ling, Shiqing, and Michael McAleer. 2003. Asymptotic theory for a vector ARMA-GARCH model. *Econometric Theory* 19: 280–310. [\[CrossRef\]](#)
- Lucchetti, Riccardo. 2002. Analytical score for multivariate GARCH models. *Computational Economics* 19: 133–43. [\[CrossRef\]](#)
- Lütkepohl, Helmut. 1996. *Handbook of Matrices*. Hoboken: John Wiley & Sons.
- Magnus, Jan R., and Heinz Neudecker. 1999. *Matrix Differential Calculus with Applications in Statistics and Econometrics*, 2nd ed. Hoboken: John Wiley.

Pelletier, Denis. 2006. Regime switching for dynamic correlations. *Journal of Econometrics* 131: 445–73.

[[CrossRef](#)]

Wang, Qian, Xianyi Zhang, Yunquan Zhang, and Qing Yi. 2013. AUGEM: Automatically generate high performance dense linear algebra kernels on x86 CPUs. Presented at the SC'13: Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis, Denver, CO, USA, 17–22 November 2013; pp. 1–12.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).