OPTIMISATION OF REAL FUNCTIONS OF COMPLEX MATRICES FOR THE ADAPTIVE ESTIMATION OF COMPLEX SOURCES

Soroush Javidi[1], *Danilo P. Mandic*[1] *and Anthony Kuh*[2]

[1]Department of Electrical and Electronic Engineering, Imperial College, London, UK
[2]Department of Electrical Engineering, University of Hawaii, Honolulu, USA
{soroush.javidi,d.mandic}@imperial.ac.uk, kuh@spectra.eng.hawaii.edu

ABSTRACT

The second order Taylor series expansion (TSE) of scalar functions of complex matrices is explored in order to provide a new tool for gradient-based optimisation in the complex domain. The expansion is provided both in the augmented real and complex matrix spaces, as well as the multidimensional complex domain. The duality (isomorphism) between the augmented spaces is established and consequently the relation of the first- and second-order terms (gradient and Hessian) of the TSE in these spaces are introduced. Finally, a study of the trade-off between performance and computational complexity of algorithms for the estimation of complex sources in the two augmented spaces is performed.

Index Terms— Complex matrix calculus, Newton optimisation, complex eigenvalues, augmented complex matrix space.

1. INTRODUCTION

The recent resurgence of interest in complex domain signal processing has been made possible by the enhanced understanding of the statistics in $\mathbb{C}$. For instance, the advances in so called augmented statistics and $\mathbb{CR}$ calculus have enabled the design and analysis of new algorithms and deeper insight into signal processing problems in $\mathbb{C}$ [1]. As algorithms based on so called augmented complex statistics are emerging, leading to more accurate but mathematically involved solutions, revisiting some aspects of complex calculus is a prerequisite to providing a set of analytic tools to support these developments.

In this direction, for real-valued functions of complex vector variables, the work by van den Bos [2] has provided a platform for modelling and optimisation via so called augmented vector spaces, with a thorough overview given in [3], where the duality between these spaces is explored. The application of these results have been recently utilised in various statistical signal processing fields, such as adaptive filtering [1].

Complex optimisation problems often involve real-valued functions of complex matrices[1]; these are a standard in communications and signal processing problems, such as in optimisation problems in MIMO systems and in blind source separation. In this paper, by complementing the work in [4], [5], we extend the concept of duality between vectors $\mathbb{R}^N$ and $\mathbb{C}^N$ in [3] to the case of complex matrix spaces, and formalise the equivalence of real-valued functions of complex matrix variables in the standard and augmented spaces up to their second-order Taylor series expansion.

We show that this is sufficient for the derivation and analysis of standard gradient-based learning algorithms. This also helps with the analysis of general signal processing algorithms in augmented matrix spaces and allows for simpler closed form solutions. Applications in Newton optimisation and blind source separation demonstrate the potential of the introduced complex matrix calculus results. This is followed by a comparison of adaptive algorithms in the real and complex matrix spaces demonstrate the trade-offs associated with the algorithms.

2. REPRESENTATIONS OF COMPLEX MATRICES

The complex matrix $\mathbf{Z} = \mathbf{Z}_r + \jmath\mathbf{Z}_i \in \mathbb{C}^{M \times N}$, with $\mathbf{Z}_r$ and $\mathbf{Z}_i$ denoting respectively the real and imaginary components, can be equivalently described as a matrix $\mathbf{\Omega}$ in the real-valued space $\mathbb{R}^{2M\times 2N}$, given by

$$\mathbf{\Omega} = \begin{bmatrix} \mathbf{Z}_r & -\mathbf{Z}_i \\ \mathbf{Z}_i & \mathbf{Z}_r \end{bmatrix} \in \mathbb{R}^{2M\times 2N} \triangleq \mathcal{R} \tag{1}$$

or as a matrix $\mathbf{V}$ in the complex conjugate-coordinate space[2] $\mathbb{C}^{2M\times 2N}$, given by

$$\mathbf{V} = \begin{bmatrix} \mathbf{Z} & \mathbf{0} \\ \mathbf{0} & \mathbf{Z}^* \end{bmatrix} \in \mathbb{C}^{2M\times 2N} \triangleq \mathcal{C} \tag{2}$$

where $\mathbf{V}$ is referred to as the *augmented* form of the complex matrix $\mathbf{Z}$ and $\mathbf{0}$ is a zero-valued matrix of size $M{\times}N$ [3]. This equivalent notation is possible due to the duality (iso-

[1] For instance the cost function in complex adaptive filtering is $E = \frac{1}{2}e(k)e^*(k)$ and is a real function of complex error $e(k)$.

[2] For simplicity, we use the notations $\mathcal{R} \triangleq \mathbb{R}^{2M\times 2N}$ and $\mathcal{C} \triangleq \mathbb{C}^{2M\times 2N}$ in the following sections.

978-1-4244-6878-2/10/$26.00 ©2010 IEEE

morphism) between the spaces $\mathcal{R}$ and $\mathcal{C}$ and is formalised by the transformation between $\mathbf{\Omega}$ and $\mathbf{V}$, described by the matrix[3]

$$\mathbf{J}_K = \begin{bmatrix} \mathbf{I} & \jmath\mathbf{I} \\ \mathbf{I} & -\jmath\mathbf{I} \end{bmatrix}. \tag{3}$$

Matrix $\mathbf{J}_K$, introduced in [2] and [3], is a square block matrix of size $2K{\times}2K$ and $\mathbf{I}$ is the identity matrix of size $K{\times}K$. The inverse of this mapping is given by

$$\mathbf{J}_K^{-1} = \frac{1}{2}\mathbf{J}_K^H \tag{4}$$

and thus matrices $\mathbf{\Omega}$ and $\mathbf{V}$ are related by

$$\mathbf{V} = \frac{1}{2}\mathbf{J}_M \mathbf{\Omega} \mathbf{J}_N^H, \qquad \mathbf{\Omega} = \frac{1}{2}\mathbf{J}_M^H \mathbf{V} \mathbf{J}_N. \tag{5}$$

Alternatively, the mapping in (5) can be written using the $\text{vec}(\cdot)$ operator[4]. In this manner[5],

$$\text{vec}(\mathbf{V}) = \frac{1}{2}(\mathbf{J}_N^* \otimes \mathbf{J}_M)\,\text{vec}(\mathbf{\Omega}) = \mathcal{J}\,\text{vec}(\mathbf{\Omega}) \tag{6}$$

$$\text{vec}(\mathbf{\Omega}) = \frac{1}{2}(\mathbf{J}_N^T \otimes \mathbf{J}_M^H)\,\text{vec}(\mathbf{V}) = \mathcal{J}^{-1}\,\text{vec}(\mathbf{V}) \tag{7}$$

and allows for a simplified and convenient method of describing the coordinate transformation, denoted by $\mathcal{J} \in \mathbb{R}^{4MN \times 4MN}$. Therefore, the Jacobian of the transformation [3] from $\mathcal{R}$ to $\mathcal{C}$ is given by

$$J_{\mathcal{C}} = \frac{\partial}{\partial \mathbf{\Omega}}\mathbf{V} = \frac{\partial\,\text{vec}(\mathbf{V})}{\partial\,\text{vec}^T(\mathbf{\Omega})} = \frac{1}{2}(\mathbf{J}_N^* \otimes \mathbf{J}_M) = \mathcal{J} \tag{8}$$

and the Jacobian of the transformation from $\mathcal{C}$ to $\mathcal{R}$ by

$$J_{\mathcal{R}} = \frac{\partial}{\partial \mathbf{V}}\mathbf{\Omega} = \frac{\partial\,\text{vec}(\mathbf{\Omega})}{\partial\,\text{vec}^T(\mathbf{V})} = \frac{1}{2}(\mathbf{J}_N^T \otimes \mathbf{J}_M^H) = \mathcal{J}^{-1}. \tag{9}$$

This illustrates that the Jacobian of the transformation $J_{\mathcal{C}}$ in (8) is equal to the coordinate transformation $\mathcal{J}$, and the Jacobian of the transformation $J_{\mathcal{R}}$ in (9) is equal to the inverse transformation $\mathcal{J}^{-1}$ [3]. As a result, the partial derivative transformations[6] between the two spaces in the vectorised format are given by

$$\frac{\partial\,\text{vec}(\cdot)}{\partial\,\text{vec}^T(\mathbf{V})} = \frac{1}{2}\frac{\partial\,\text{vec}(\cdot)}{\partial\,\text{vec}^T(\mathbf{\Omega})}(\mathbf{J}_N^T \otimes \mathbf{J}_M^H) \tag{10}$$

$$\frac{\partial\,\text{vec}(\cdot)}{\partial\,\text{vec}^T(\mathbf{\Omega})} = \frac{1}{2}\frac{\partial\,\text{vec}(\cdot)}{\partial\,\text{vec}^T(\mathbf{V})}(\mathbf{J}_N^* \otimes \mathbf{J}_M) \tag{11}$$

and are row vectors of size $1{\times}4MN$. Note that the partial derivative is defined as a row operator [3] with the transpose notation $\frac{\partial\,\text{vec}(\cdot)}{\partial\,\text{vec}^T(\cdot)}$ used to emphasise this fact.

For a real-valued scalar function of vector complex variables $f(\mathbf{Z}, \mathbf{Z}^*) : \mathbb{C}^{M\times N} \times \mathbb{C}^{M\times N} \mapsto \mathbb{R}$, the partial derivative transforms can be simplified to an equivalent form [5]

$$\frac{\partial f}{\partial \mathbf{V}} = \frac{1}{2}\mathbf{J}_N \frac{\partial f}{\partial \mathbf{\Omega}} \mathbf{J}_M^H \tag{12}$$

$$\frac{\partial f}{\partial \mathbf{\Omega}} = \frac{1}{2}\mathbf{J}_N^H \frac{\partial f}{\partial \mathbf{V}} \mathbf{J}_M \tag{13}$$

where $\frac{\partial f}{\partial \mathbf{V}}$ and $\frac{\partial f}{\partial \mathbf{\Omega}}$ are matrices of size $2N{\times}2M$. The proof for this alternative form is given in the next section, and follows directly from the first order expansion of $f(\mathbf{Z}, \mathbf{Z}^*)$. Also, note that $\frac{\partial(\cdot)}{\partial \mathbf{V}}$ and $\frac{\partial(\cdot)}{\partial \mathbf{\Omega}}$ are shorthand notations and are calculated as

$$\frac{\partial(\cdot)}{\partial \mathbf{V}} = \begin{bmatrix} \frac{\partial(\cdot)}{\partial \mathbf{Z}} & \mathbf{0} \\ \mathbf{0} & \frac{\partial(\cdot)}{\partial \mathbf{Z}^*} \end{bmatrix}^T, \quad \frac{\partial(\cdot)}{\partial \mathbf{\Omega}} = \begin{bmatrix} \frac{\partial(\cdot)}{\partial \mathbf{Z}_r} & -\frac{\partial(\cdot)}{\partial \mathbf{Z}_i} \\ \frac{\partial(\cdot)}{\partial \mathbf{Z}_i} & \frac{\partial(\cdot)}{\partial \mathbf{Z}_\mathbf{r}} \end{bmatrix}^T.$$

The real-valued scalar function f can be equivalently described in terms of coordinates in either $\mathbb{C}^{M\times N}$, $\mathcal{R}$ and $\mathcal{C}$. Following on [3], the TSE of the function $f(\mathbf{\Omega})$ up to the second term is

$$f(\mathbf{\Omega} + \Delta\mathbf{\Omega}) = f(\mathbf{\Omega}) + \text{Tr}\Big(\frac{\partial f}{\partial \mathbf{\Omega}}\Delta\mathbf{\Omega}\Big) + \frac{1}{2}\,\text{vec}^T(\Delta\mathbf{\Omega})\mathbf{H}_{\Omega\Omega}\,\text{vec}(\Delta\mathbf{\Omega}) \tag{14}$$

where symbol $\text{Tr}(\cdot)$ denotes the matrix trace operator, $\Delta\mathbf{\Omega}$ and $\Delta\mathbf{V}$ are of the form given in (1) and (2), and $\mathbf{H}_{\Omega\Omega}$ is a real valued Hessian matrix given by

$$\mathbf{H}_{\Omega\Omega} = \frac{\partial}{\partial\,\text{vec}^T(\mathbf{\Omega})}\,\text{vec}\left(\left[\frac{\partial f}{\partial \mathbf{\Omega}}\right]^T\right) \in \mathbb{R}^{4MN\times 4MN}. \tag{15}$$

2.1. Duality of First-Order Taylor Series Expansions

Upon rewriting the first-order expansion term in (14) in the vectorised format, and using (7) and (10), we have

$$\begin{aligned}\text{Tr}\Big(\frac{\partial f}{\partial \mathbf{\Omega}}\Delta\mathbf{\Omega}\Big) &= \frac{\partial f}{\partial\,\text{vec}^T(\mathbf{\Omega})}\,\text{vec}(\Delta\mathbf{\Omega}) \\ &= \frac{1}{2}\frac{\partial f}{\partial\,\text{vec}^T(\mathbf{\Omega})}(\mathbf{J}_N^T \otimes \mathbf{J}_M^H)\,\text{vec}(\Delta\mathbf{V}) \\ &= \frac{\partial f}{\partial\,\text{vec}^T(\mathbf{V})}\,\text{vec}(\Delta\mathbf{V}) = \text{Tr}\Big(\frac{\partial f}{\partial \mathbf{V}}\Delta\mathbf{V}\Big)\end{aligned} \tag{16}$$

which is the first-order TSE of $f(\mathbf{V})$ in $\mathcal{C}$. Furthermore, using the relations (5), we have

$$\text{Tr}\Big(\frac{\partial f}{\partial \mathbf{\Omega}}\Delta\mathbf{\Omega}\Big) = \text{Tr}\Big(\frac{1}{2}\frac{\partial f}{\partial \mathbf{\Omega}}\mathbf{J}_M^H(\Delta\mathbf{V})\mathbf{J}_N\Big)$$

$$\text{Tr}\Big(\frac{\partial f}{\partial \mathbf{V}}\Delta\mathbf{V}\Big) = \text{Tr}\Big(\frac{1}{2}\frac{\partial f}{\partial \mathbf{V}}\mathbf{J}_M(\Delta\mathbf{\Omega})\mathbf{J}_N^H\Big)$$

and due to the duality between $\mathcal{R}$ and $\mathcal{C}$, and the equivalence in the first-order terms in the corresponding TSEs we have[7]

$$\begin{aligned}\text{Tr}\Big(\frac{\partial f}{\partial \mathbf{\Omega}}\Delta\mathbf{\Omega}\Big) &= \text{Tr}\Big(\frac{1}{2}\frac{\partial f}{\partial \mathbf{V}}\mathbf{J}_M(\Delta\mathbf{\Omega})\mathbf{J}_N^H\Big) \\ &= \text{Tr}\Big(\frac{1}{2}\mathbf{J}_N^H\frac{\partial f}{\partial \mathbf{V}}\mathbf{J}_M(\Delta\mathbf{\Omega})\Big)\end{aligned} \tag{17}$$

[3] Alternatively, by using the scaling factor $1/\sqrt{2}$ in the definition in (3), the matrix $\mathbf{J}$ becomes a unitary matrix [6].

[4] The vec operator stacks the columns of a matrix into a single column in a chronological order [4].

[5] The vec operator and Kronecker product $\otimes$ are related by $\text{vec}(\mathbf{RQS}) = (\mathbf{S}^T \otimes \mathbf{R})\,\text{vec}(\mathbf{Q})$.

[6] Also termed the cogradient transformations in [3].

[7] We also make use of the identity $\text{Tr}(\mathbf{RQ}) = \text{Tr}(\mathbf{QR})$.

and

$$\mathrm{Tr}\Big(\frac{\partial f}{\partial \mathbf{V}}\Delta\mathbf{V}\Big) = \mathrm{Tr}\Big(\frac{1}{2}\frac{\partial f}{\partial \mathbf{\Omega}}\mathbf{J}_M^H(\Delta\mathbf{V})\mathbf{J}_N\Big)$$
$$= \mathrm{Tr}\Big(\frac{1}{2}\mathbf{J}_N\frac{\partial f}{\partial \mathbf{\Omega}}\mathbf{J}_M^H(\Delta\mathbf{V})\Big). \quad (18)$$

The equivalence of the terms on both sides of relations (17) and (18) results in the simplified partial derivative transforms given in (12) and (13).

Now, to produce the first-order expansion of $f(\mathbf{Z})$ in $\mathbb{C}^{M\times N}$, we can expand the first-order terms of $f(\mathbf{V})$ to yield

$$\mathrm{Tr}\Big(\frac{\partial f}{\partial \mathbf{V}}\Delta\mathbf{V}\Big) = \mathrm{Tr}\left(\Big(\frac{\partial f}{\partial \mathbf{Z}}\Big)^T\Delta\mathbf{Z} + \Big(\frac{\partial f}{\partial \mathbf{Z}^*}\Big)^T\Delta\mathbf{Z}^*\right)$$
$$= 2\Re\left\{\mathrm{Tr}\left(\Big(\frac{\partial f}{\partial \mathbf{Z}}\Big)^T\Delta\mathbf{Z}\right)\right\} \quad (19)$$

where $\frac{\partial f}{\partial \mathbf{Z}^*} = (\frac{\partial f}{\partial \mathbf{Z}})^*$, as $f \in \mathbb{R}$. Also note that the gradient in the direction of steepest descent is given by $\frac{\partial f}{\partial \mathbf{Z}^*}$ [4, 5].

2.2. Eigenvalue Analysis of Hessian Matrices

We shall now establish the relationships between second-order terms in the TSE of a scalar f in the spaces $\mathbb{C}^{M\times N}$, $\mathcal{R}$ and $\mathcal{C}$. In addition, by analysing the relationship between the Hessian matrices in $\mathcal{R}$ and $\mathcal{C}$, a relation between the eigenvalues of the corresponding Hessian matrices is provided.

Observe the relationship between the real Hessian matrix $\mathbf{H}_{\Omega\Omega}$ in (15) and the complex Hessian matrix $\mathbf{H}_{VV}$, given by[8]

$$\mathbf{H}_{VV} = \frac{\partial}{\partial\,\mathrm{vec}^T(\mathbf{V})}\,\mathrm{vec}\left(\left[\frac{\partial f}{\partial \mathbf{V}}\right]^H\right) \in \mathbb{C}^{4MN\times 4MN}. \quad (20)$$

From (15), we have[9]

$$\mathbf{H}_{\Omega\Omega} = \frac{\partial}{\partial\,\mathrm{vec}^T(\mathbf{\Omega})}\,\mathrm{vec}\left(\left[\frac{\partial f}{\partial \mathbf{\Omega}}\right]^H\right)$$
$$= \frac{\partial}{\partial\,\mathrm{vec}^T(\mathbf{\Omega})}\left\{\mathrm{vec}\left(\frac{1}{2}\Big(\mathbf{J}_N^H\frac{\partial f}{\partial \mathbf{V}}\mathbf{J}_M\Big)^H\right)\right\}$$
$$= \frac{\partial}{\partial\,\mathrm{vec}^T(\mathbf{V})}\left\{\frac{1}{2}(\mathbf{J}_N^T\otimes\mathbf{J}_M^H)\,\mathrm{vec}\Big(\frac{\partial f}{\partial \mathbf{V}}\Big)^H\right\}\frac{1}{2}(\mathbf{J}_N^*\otimes\mathbf{J}_M)$$
$$= \frac{1}{4}(\mathbf{J}_N^T\otimes\mathbf{J}_M^H)\frac{\partial}{\partial\,\mathrm{vec}^T(\mathbf{V})}\,\mathrm{vec}\Big(\frac{\partial f}{\partial \mathbf{V}}\Big)^H(\mathbf{J}_N^*\otimes\mathbf{J}_M)$$
$$= \frac{1}{4}(\mathbf{J}_N^T\otimes\mathbf{J}_M^H)\mathbf{H}_{VV}(\mathbf{J}_N^*\otimes\mathbf{J}_M) \quad (21)$$

which is the relationship between real and complex Hessian matrices, written in terms of $\mathbf{H}_{VV}$. This relationship can also be expressed in terms of the real Hessian matrix $\mathbf{H}_{\Omega\Omega}$ by noticing that the two Kronecker product terms are the inverse of one another[10]. Thus

$$\mathbf{H}_{VV} = \frac{1}{4}(\mathbf{J}_N^*\otimes\mathbf{J}_M)\mathbf{H}_{\Omega\Omega}(\mathbf{J}_N^T\otimes\mathbf{J}_M^H). \quad (22)$$

The analysis of the eigenvalues of the two Hessian matrices will assist in understanding their duality. Following the approach in [2] and [3], consider the linear system

$$(\mathbf{H}_{VV} - \lambda^V\mathbf{I})\mathbf{u} = \mathbf{0} \quad \Rightarrow \quad (\mathbf{H}_{VV} - \lambda^V\mathbf{I}) = \mathbf{0} \quad (23)$$

where the set of solutions spans the eigenspace. Using the relation (22) we have

$$\mathbf{H}_{VV} - \lambda^V\mathbf{I} = \frac{1}{4}(\mathbf{J}_N^*\otimes\mathbf{J}_M)\mathbf{H}_{\Omega\Omega}(\mathbf{J}_N^T\otimes\mathbf{J}_M^H) \quad (24)$$
$$- \lambda^V\frac{1}{4}(\mathbf{J}_N^*\otimes\mathbf{J}_M)(\mathbf{J}_N^T\otimes\mathbf{J}_M^H)$$
$$= \frac{1}{4}(\mathbf{J}_N^*\otimes\mathbf{J}_M)\underbrace{[\mathbf{H}_{\Omega\Omega} - \lambda^V\mathbf{I}]}_{\Rightarrow\lambda^V=\lambda^\Omega}(\mathbf{J}_N^T\otimes\mathbf{J}_M^H)$$

where $\{\lambda^V\}$ are the eigenvalues of the complex Hessian matrix. This demonstrates that for every eigenvalue λ^V of the complex-valued Hessian matrix $\mathbf{H}_{VV}$, there is a corresponding eigenvalue λ^Ω of the real-valued Hessian matrix $\mathbf{H}_{\Omega\Omega}$, and that these eigenvalues are equal

$$\lambda^\Omega = \lambda^V. \quad (25)$$

2.3. Duality of Second-Order Taylor Series Expansions

This section effectively extends the analysis for the vector case presented in [3]. The second-order expansion term in $\mathcal{C}$ is obtained from (14) using the relationship (21) such that

$$\frac{1}{2}\,\mathrm{vec}^T(\Delta\mathbf{\Omega})\mathbf{H}_{\Omega\Omega}\,\mathrm{vec}(\Delta\mathbf{\Omega}) = \frac{1}{2}\,\mathrm{vec}^H(\Delta\mathbf{\Omega})\mathbf{H}_{\Omega\Omega}\,\mathrm{vec}(\Delta\mathbf{\Omega})$$
$$= \frac{1}{2}\big(\frac{1}{2}\,\mathrm{vec}^H(\Delta\mathbf{V})(\mathbf{J}_N^*\otimes\mathbf{J}_M)\big)\mathbf{H}_{\Omega\Omega}\big(\frac{1}{2}(\mathbf{J}_N^T\otimes\mathbf{J}_M^H)\,\mathrm{vec}(\Delta\mathbf{V})\big)$$
$$= \frac{1}{2}\,\mathrm{vec}^H(\Delta\mathbf{V})\mathbf{H}_{VV}\,\mathrm{vec}(\Delta\mathbf{V}). \quad (26)$$

The components of the second-order expansions in $\mathcal{C}$ can now be written in terms of matrix $\mathbf{Z}$ to derive the second-order expansion in the standard $\mathbb{C}^{M\times N}$ space, that is

$$\frac{1}{2}\,\mathrm{vec}^H(\Delta\mathbf{V})\mathbf{H}_{VV}\,\mathrm{vec}(\Delta\mathbf{V}) = \quad (27)$$
$$\Re\Big\{\mathrm{vec}^H(\Delta\mathbf{Z})\mathbf{H}_{ZZ}\,\mathrm{vec}(\Delta\mathbf{Z}) + \mathrm{vec}^H(\Delta\mathbf{Z})\mathbf{H}_{Z^*Z}\,\mathrm{vec}^*(\Delta\mathbf{Z})\Big\}$$

where the operator $\Re\{\cdot\}$ denotes the real part of a complex variable, $\mathbf{H}_{ZZ} \triangleq \frac{\partial\,\mathrm{vec}(\partial f/\partial\mathbf{Z})^*}{\partial\,\mathrm{vec}^T(\mathbf{Z})}$ and $\mathbf{H}_{Z^*Z} \triangleq \frac{\partial\,\mathrm{vec}(\partial f/\partial\mathbf{Z})^*}{\partial\,\mathrm{vec}^T(\mathbf{Z}^*)}$. To summarise, the expansion of f in $\mathcal{R}$ is illustrated in (14), whereas the expansion in $\mathcal{C}$ is shown through the isomorphism

[8] The notation $\mathrm{vec}([\cdot]^T)$ is used interchangeably with $\mathrm{vec}(\cdot)^T$. Note the difference from $\mathrm{vec}^T(\cdot)$.

[9] Notice that since $\mathbf{H}_{\Omega\Omega}$ in (15) is real-valued, for convenience we can apply the complex conjugate operator to both sides of (15) and hence replace $(\cdot)^T$ by $(\cdot)^H$.

[10] This can be observed from (8) and (9). Alternatively, the identity $(\mathbf{R}\otimes\mathbf{Q})^{-1} = \mathbf{R}^{-1}\otimes\mathbf{Q}^{-1}$ and (4) can be used to obtain the same result, i.e. $\frac{1}{4}(\mathbf{J}_N^*\otimes\mathbf{J}_M)(\mathbf{J}_N^T\otimes\mathbf{J}_M^H) = \mathbf{I}$.

between the two spaces given in (16) and (26), to yield

$$f(\mathbf{V}+\Delta\mathbf{V}) = f(\mathbf{V}) + \mathrm{Tr}\Big(\frac{\partial f}{\partial \mathbf{V}}\Delta\mathbf{V}\Big) + \frac{1}{2}\,\mathrm{vec}^H(\Delta\mathbf{V})\mathbf{H}_{VV}\,\mathrm{vec}(\Delta\mathbf{V}) \qquad (28)$$

Similarly, the TSE of a scalar function of complex matrix variables f in $\mathbb{C}^{M\times N}$ is given by (19) for the first term, and in (27) for the second term, that is

$$f(\mathbf{Z}+\Delta\mathbf{Z}) = f(\mathbf{Z}) + 2\Re\Big\{\mathrm{Tr}\Big(\Big(\frac{\partial f}{\partial \mathbf{Z}}\Big)^T\Delta\mathbf{Z}\Big)\Big\} \qquad (29)$$

$$+\,\Re\Big\{\mathrm{vec}^H(\Delta\mathbf{Z})\mathbf{H}_{ZZ}\,\mathrm{vec}(\Delta\mathbf{Z}) + \mathrm{vec}^H(\Delta\mathbf{Z})\mathbf{H}_{Z^*Z}\,\mathrm{vec}^*(\Delta\mathbf{Z})\Big\}$$

3. APPLICATION EXAMPLES

To illustrate the potential of the derived results, we shall consider two case studies: Newton optimisation and Blind Source Separation. Due to lack of space, only a concise implementation of the theory is provided.

3.1. Optimisation in the Augmented Matrix Spaces

A classic optimisation application, illustrated in [2], is the minimisation of the real-valued function $f : \mathbb{C}^N\times\mathbb{C}^N \mapsto \mathbb{R}$ using the Newton method. We will consider the extension of this approach to functions of complex matrices $f : \mathbb{C}^{M\times N}\times\mathbb{C}^{M\times N} \mapsto \mathbb{R}$, to calculate the minima $\partial f/\partial\mathbf{\Omega} = \mathbf{0}$ and $\partial f/\partial\mathbf{V} = \mathbf{0}$. By taking the derivative of the second order expansion term of $f(\mathbf{\Omega})$ in (14), and $f(\mathbf{V})$, in (28), and equating to zero, we have

$$\mathbf{H}_{\Omega\Omega}\,\mathrm{vec}(\Delta\mathbf{\Omega}) = -\Big(\frac{\partial f}{\partial\,\mathrm{vec}^T(\mathbf{\Omega})}\Big)^T \qquad (30)$$

$$\mathbf{H}_{VV}\,\mathrm{vec}(\Delta\mathbf{V}) = -\Big(\frac{\partial f}{\partial\,\mathrm{vec}^T(\mathbf{V})}\Big)^H. \qquad (31)$$

The benefit of this formulation is that it allows complex optimisation problems to be cast in augmented matrix spaces, which when combined with $\mathbb{CR}$ calculus, provide a simpler and easier to understand way of calculating the optimal solution.

3.2. Derivative Calculation in Blind Source Separation

In the derivation of the complex blind source separation algorithm based on maximum likelihood, it is necessary to calculate the derivative $\frac{\partial \log|\det(\mathbf{\Omega})|}{\partial\mathbf{Z}^*}$. The method provided in [7] requires the introduction of a new symmetric matrix and further algebraic manipulation. A more straightforward calculation, based on the introduced framework, gives

$$\begin{aligned}\log|\det(\mathbf{\Omega})| &= \log|\det(\tfrac{1}{2}\mathbf{J}^H\mathbf{V}\mathbf{J})|\\ &= \log|\det(\tfrac{1}{2}\mathbf{J}^H)\det(\mathbf{V})\det(\mathbf{J})|\\ &= \log|\det(\mathbf{V})| = \log|\det(\mathbf{Z})\cdot\det(\mathbf{Z}^*)|\\ &= \log|\det(\mathbf{Z})| + \log|\det(\mathbf{Z}^*)|\end{aligned}$$

and therefore

$$\frac{\partial\log|\det(\mathbf{\Omega})|}{\partial\mathbf{Z}^*} = \Big[\frac{\partial\log|\det(\mathbf{Z})|}{\partial\mathbf{Z}} + \frac{\partial\log|\det(\mathbf{Z}^*)|}{\partial\mathbf{Z}}\Big]^* = \mathbf{Z}^{-H}$$

where we have also used some fundamental results from linear algebra [8] and matrix derivatives [5].

4. NORM-BASED COST FUNCTIONS

Several cost functions encountered in signal processing research are defined based on matrix inputs [4]. Here we shall address on norm-based cost functions $\mathcal{Q}(\mathbf{A},\mathbf{A}^*) : \mathbb{C}^{N\times N}\times\mathbb{C}^{N\times N} \mapsto \mathbb{R}$ given by

$$\mathcal{Q}(\mathbf{A},\mathbf{A}^*) = \|\mathbf{A}\|_F^2 = \mathrm{Tr}(\mathbf{A}^H\mathbf{A}) \qquad (32)$$

where $\|\cdot\|_F$ denotes the Frobenius norm. Consider the linear predictor of $\mathbf{U}$ given by

$$\hat{\mathbf{U}} = \mathbf{W}^T\mathbf{Z}, \qquad (33)$$

with estimation error $\mathbf{E} = \mathbf{U} - \hat{\mathbf{U}}$, input matrix $\mathbf{Z}$ and weight matrix $\mathbf{W} \in \mathbb{C}^{N\times N}$, and the norm-based cost function $\mathcal{Q} = \|\mathbf{E}\|_F^2 = \mathrm{Tr}\{\mathbf{E}^H\mathbf{E}\}$. The optimal value of $\mathbf{W}$ can be obtained adaptively using a gradient descent method that minimises the cost function. Thus using $\mathbb{CR}$ calculus

$$\mathbf{W}_{k+1} = \mathbf{W}_k - \mu\nabla_{\mathbf{W}_k}\mathcal{Q} = \mathbf{W}_k + \mu\mathbf{E}_k\mathbf{Z}_k^* \qquad (34)$$

which will be referred to as the block complex least mean square (b-CLMS) algorithm, where μ is the step-size. Alternatively, by assuming a widely linear model [9] of $\mathbf{U}$ based on the input $\mathbf{Z}$ and its conjugate $\mathbf{Z}^*$, the output of the widely linear predictor is

$$\hat{\mathbf{U}}_{WL} = \mathbf{W}^T\mathbf{Z} + \mathbf{V}^T\mathbf{Z}^* \qquad (35)$$

and $\mathbf{W}$ and $\mathbf{V}$ are the complex $N\times N$ weight matrices. The cost function can be minimised for both matrices to achieve the gradient descent algorithms[11]

$$\begin{aligned}\mathbf{W}_{k+1} &= \mathbf{W}_k + \eta\mathbf{E}_k\mathbf{Z}_k^*\\ \mathbf{V}_{k+1} &= \mathbf{V}_k + \eta\mathbf{E}_k\mathbf{Z}_k\end{aligned} \qquad (36)$$

and η is the step-size. We will refer to (36) as the block augmented complex least mean square (b-ACLMS) algorithm. Now consider the matrix analog of the dual channel real least mean square (DCRLMS) algorithm described in [10], with real-valued input/output relation

$$\begin{bmatrix}\hat{\mathbf{Y}}_1\\ \hat{\mathbf{Y}}_2\end{bmatrix} = \begin{bmatrix}\mathbf{H}_{11} & \mathbf{H}_{12}\\ \mathbf{H}_{21} & \mathbf{H}_{22}\end{bmatrix}^T\begin{bmatrix}\mathbf{X}_1\\ \mathbf{X}_2\end{bmatrix} \qquad (37)$$

where $\mathbf{X}_i$ are the real-valued input matrices, $\hat{\mathbf{Y}}_i$ are the estimated output. The matrix of weight matrices $\mathbf{H}_{pq} \in \mathbb{R}^{N\times N}$ is updated adaptively as

$$\mathbf{H}_{pq,k+1} = \mathbf{H}_{pq,k} + \rho\mathbf{E}_q\mathbf{X}_{p,k}, \quad p,q = \{1,2\} \qquad (38)$$

[11] See [1] for more details.

and $\mathbf{E}_{q,k} = \mathbf{Y}_{q,k} - \hat{\mathbf{Y}}_{q,k}$ is the estimation error and ρ is the step-size. We will refer to the update algorithms (38) as block DCRLMS (b-DCRLMS).

In order to perform analysis between the update algorithms in $\mathbb{C}^{N\times N}$ and $\mathbb{R}^{N\times N}$, we will write the linear input relation (33) in terms of its real and imaginary components $\mathbf{U}^r$ and $\mathbf{U}^i$, to obtain

$$\hat{\mathbf{U}}^r = \mathbf{W}^{r^T}\mathbf{Z}^r - \mathbf{W}^{i^T}\mathbf{Z}^i, \quad \hat{\mathbf{U}}^i = \mathbf{W}^{i^T}\mathbf{Z}^r + \mathbf{W}^{r^T}\mathbf{Z}^i \quad (39)$$

and for the widely linear relation (35), we have

$$\hat{\mathbf{U}}^r_{WL} = (\mathbf{W}^r + \mathbf{V}^r)^T\mathbf{Z}^r + (\mathbf{V}^i - \mathbf{W}^i)^T\mathbf{Z}^i \quad (40)$$

$$\hat{\mathbf{U}}^i_{WL} = (\mathbf{W}^i + \mathbf{V}^i)^T\mathbf{Z}^r + (\mathbf{W}^r - \mathbf{V}^i)^T\mathbf{Z}^i. \quad (41)$$

Similarly, the update algorithms can be written in terms of the updates for the real and imaginary components of the weight matrices. For the b-CLMS algorithm (34), we thus have

$$\mathbf{W}^r_{k+1} = \mathbf{W}^r_k + \mu(\mathbf{E}^r_k\mathbf{Z}^r_k + \mathbf{E}^i_k\mathbf{Z}^i_k) \quad (42)$$

$$\mathbf{W}^i_{k+1} = \mathbf{W}^i_k + \mu(\mathbf{E}^i_k\mathbf{Z}^r_k - \mathbf{E}^r_k\mathbf{Z}^i_k), \quad (43)$$

while for the b-ACLMS algorithm (36)

$$\mathbf{W}^r_{k+1} = \mathbf{W}^r_k + \eta(\mathbf{E}^r_k\mathbf{Z}^r_k + \mathbf{E}^i_k\mathbf{Z}^i_k) \quad (44)$$

$$\mathbf{W}^i_{k+1} = \mathbf{W}^i_k + \eta(\mathbf{E}^i_k\mathbf{Z}^r_k - \mathbf{E}^r_k\mathbf{Z}^i_k) \quad (45)$$

$$\mathbf{V}^r_{k+1} = \mathbf{V}^r_k + \eta(\mathbf{E}^r_k\mathbf{Z}^r_k - \mathbf{E}^i_k\mathbf{Z}^i_k) \quad (46)$$

$$\mathbf{V}^i_{k+1} = \mathbf{V}^i_k + \eta(\mathbf{E}^i_k\mathbf{Z}^r_k + \mathbf{E}^r_k\mathbf{Z}^i_k). \quad (47)$$

4.1. Adaptive Linear Algorithms

To compare the input/output relation and the dynamics of the b-CLMS and b-DRCRLMS algorithms, for the same inputs from (39) and (37) we have

$$\mathbf{X}_1 = \mathbf{Z}^r, \quad \mathbf{X}_2 = \mathbf{Z}^i \quad (48)$$

and the corresponding errors are defined so that

$$\mathbf{E}_1 = \mathbf{E}^r, \quad \mathbf{E}_2 = \mathbf{E}^i. \quad (49)$$

Thus, for the same outputs $\hat{\mathbf{Y}}_1 = \hat{\mathbf{U}}^r$ and $\hat{\mathbf{Y}}_2 = \hat{\mathbf{U}}^i$, we have

$$\begin{aligned} \mathbf{H}_{11} &= \mathbf{W}^r & \mathbf{H}_{12} &= \mathbf{W}^i \\ \mathbf{H}_{21} &= -\mathbf{W}^i & \mathbf{H}_{22} &= \mathbf{W}^r \end{aligned} \quad (50)$$

It is clear that the b-CLMS input/output relation is a constrained version of the b-DCRLMS, where fixed values are assigned to the $\mathbf{H}_{ij}$ matrices.

The dynamic behaviour of the two update algorithms can be readily compared from (38) and (42), illustrating that the two algorithms are not equivalent, due to the different dynamics of the updates in $\mathbb{C}^{N\times N}$ and $\mathbb{R}^{N\times N}$. Also notice that while the updates $\Delta\mathbf{W}^r_k$ and $\Delta\mathbf{W}^i_k$ of the b-CLMS algorithm depend on both the real and imaginary error components, the b-DCLMS update $\Delta\mathbf{H}_{ij}$ is calculated based on only the error from one channel. However, by assuming the constraints (50) on the weights $\mathbf{H}_{ij}$, we can deduce that

$$\Delta\mathbf{H}_{11,k} = \Delta\mathbf{H}_{22,k} = \frac{1}{2}(\mathbf{E}_{1,k}\mathbf{X}_{1,k} + \mathbf{E}_{2,k}\mathbf{X}_{2,k}) = \frac{1}{2}\Delta\mathbf{W}^r_k$$

$$\Delta\mathbf{H}_{12,k} = -\Delta\mathbf{H}_{21,k} = \frac{1}{2}(\mathbf{E}_{2,k}\mathbf{X}_{1,k} - \mathbf{E}_{1,k}\mathbf{X}_{2,k}) = \frac{1}{2}\Delta\mathbf{W}^i_k \quad (51)$$

and so for as equal step-size $\rho = \mu$, the b-DCRLMS algorithm converges to the optimal solution two times slower as the b-CLMS algorithm.

4.2. Adaptive Widely Linear Algorithms

We shall now compare the input/output relation of the widely linear model (35) to the dual channel real-valued model in (37). Assuming the same input relations (48) and by matching the output errors (49), the component expansions in (40)–(41) provide the relation between the corresponding outputs, such that

$$\begin{aligned} \mathbf{H}_{11} &= (\mathbf{W}^r + \mathbf{V}^r) & \mathbf{H}_{12} &= (\mathbf{W}^i + \mathbf{V}^i) \\ \mathbf{H}_{21} &= (\mathbf{V}^i - \mathbf{W}^i) & \mathbf{H}_{22} &= (\mathbf{W}^r - \mathbf{V}^r) \end{aligned} \quad (52)$$

result in the equivalent outputs $\hat{\mathbf{Y}}_1 = \mathbf{U}^r_{WL}$ and $\hat{\mathbf{Y}}_2 = \mathbf{U}^i_{WL}$. We can then establish the relationship between the dynamics of the b-ACLMS and b-DCRLMS algorithms through simple algebraic manipulations of (44)–(47), where for the same step-size $\rho = \eta$ we have the equivalence

$$\Delta\mathbf{H}_{11,k} = \mathbf{E}_{1,k}\mathbf{X}_{1,k} = \frac{1}{2}(\Delta\mathbf{W}^r_k + \Delta\mathbf{V}^r_k)$$

$$\Delta\mathbf{H}_{12,k} = \mathbf{E}_{2,k}\mathbf{X}_{1,k} = \frac{1}{2}(\Delta\mathbf{W}^i_k + \Delta\mathbf{V}^i_k)$$

$$\Delta\mathbf{H}_{21,k} = \mathbf{E}_{1,k}\mathbf{X}_{2,k} = \frac{1}{2}(\Delta\mathbf{V}^i_k - \Delta\mathbf{W}^i_k)$$

$$\Delta\mathbf{H}_{22,k} = \mathbf{E}_{2,k}\mathbf{X}_{2,k} = \frac{1}{2}(\Delta\mathbf{W}^r_k - \Delta\mathbf{V}^r_k). \quad (53)$$

Therefore, the b-DCRLMS is the real-valued equivalent of the b-ACLMS algorithm, while having a convergence rate twice as slow as that of its complex counterpart. However, due to its design based on the optimisation of a widely linear model, the b-ACLMS is better suited for modelling of complex data as it is optimal for both second order circular and noncircular signals. Finally, note that these results are in line with the existing results on adaptive algorithms in $\mathbb{R}^N$ and $\mathbb{C}$ [1].

4.3. Computational Complexity of Adaptive Algorithms

In practical applications, the overheads associated with the implementation of real, complex and augmented complex algorithms are often equally important as their performance, and it is important to address a trade-off between the accuracy and performance, in order to reduce power consumption. To compare the computational complexity of the b-CLMS, b-ACLMS and b-DCRLMS algorithms, the measurement used was the 'flop', defined as the number of floating point operations [11]. Table 4.3 states the number of flops for each

Table 1. Computational complexity of the real- and complex-valued adaptive algorithms. The variable N denotes the size of a square matrix.

Alg.	b-CLMS	b-ACLMS	b-DCRLMS
Flops	$2(3N^2 + 4N^3)$	$4(3N^2 + 4N^3)$	$4(3N^2 + 4N^3)$

adaptive algorithm, where N is the length of a square matrix while Figure 1 illustrates the increase in the computational complexity for an increase in the size of the data matrix for the respective algorithms.

It can be seen that while the computational complexity of the b-CLMS and b-DCRLMS algorithms are similar, the b-ACLMS algorithm has a higher computational cost for the same matrix size[12]. Likewise, for data matrices of size $N \geq 10$, the cost of computation becomes an important factor, while for $N < 10$, the number of flops are approximately the same across all algorithms and we focus on the performance of the algorithm. Given the equivalence of the b-ACLMS and b-DCRLMS algorithms, the implementation of the b-ACLMS is obviously less computationally effective than that of the b-DCRLMS, while providing a natural processing environment for complex data.

5. CONCLUSIONS

The second order Taylor seris expansion (TSE) of real-valued scalar functions of complex matrices in $\mathbb{C}^{M \times N}$, and augmented real $\mathcal{R}$ and complex $\mathcal{C}$ spaces has been analysed. The duality between the calculation in the augmented real and complex space is demonstrated by making use of the isomorphism between the two spaces. The practicality and efficiency of the proposed approach has been demonstrated on two signal processing applications, and the trade-off between the performance and computational cost has been highlighted.

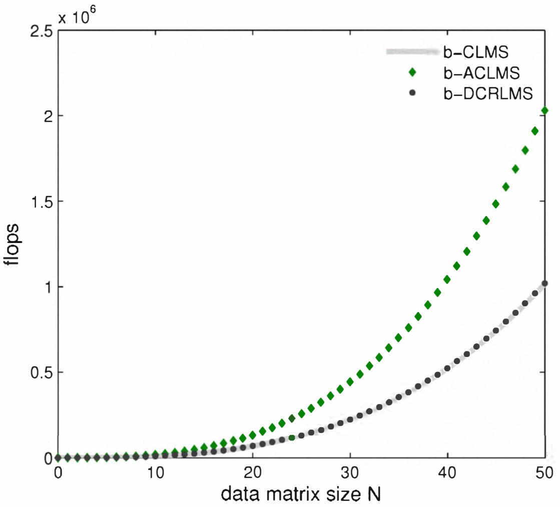

Fig. 1. Computational complexity of the b-CLMS, b-ACLMS and b-DCRLMS algorithms

6. REFERENCES

[1] D. P. Mandic and S. L. Goh, *Complex Valued Nonlinear Adaptive Filters: Noncircularity, Widely Linear and Neural Models*, Wiley, 2009.

[2] A. van den Bos, "Complex gradient and Hessian," *IEE Proceedings of Vision, Image and Signal Processing*, vol. 141, no. 6, pp. 380–383, 1994.

[3] K. Kreutz-Delgado, "The complex gradient operator and the $\mathbb{CR}$-calculus," *Dept. of Electrical and Computer Engineering, UC San Diego, Course Lecture Supplement No. ECE275A*, pp. 1–74, 2006.

[4] A. Hjørungnes and D. Gesbert, "Complex-valued matrix differentiation: Techniques and key results," *IEEE Transactions on Signal Processing*, vol. 55, no. 6, pp. 2740–2746, 2007.

[5] A. Hjørungnes, D. Gesbert, and D. P. Palomar, "Unified theory of complex-valued matrix differentiation," in *ICASSP 2007*, 2007, vol. 3, pp. 345–348.

[6] P.J. Schreier and L.L. Scharf, "Second-order analysis of improper complex random vectors and processes," *IEEE Transactions on Signal Processing*, vol. 51, no. 3, pp. 714–725, 2003.

[7] T. Adalı and H. Li, "A practical formulation for computation of complex gradients and its application to maximum likelihood," in *ICASSP 2007*, 2007, pp. 633–636.

[8] R. A. Horn and C. A. Johnson, *Matrix Analysis*, Cambridge University Press, 1985.

[9] B. Picinbono and P. Chevalier, "Widely linear estimation with complex data," *IEEE Transactions on Signal Processing*, vol. 43, no. 8, pp. 2030–2033, 1995.

[10] Y. Huang and J. Benesty, *Audio signal processing for next-generation multimedia communication systems*, Springer, 2004.

[11] G. H. Golub and C. F. Van Loan, *Matrix computations*, Johns Hopkins University Press, 1996.

[12] The b-DCRLMS algorithm has an additional overhead of $\mathcal{O}(N^2)$ with $2N^2$ flops compared to the b-CLMS algorithm, while the extra computational complexity of the b-ACLMS compared to the b-DCRLMS is $\mathcal{O}(N^3)$, that is, $4N^2 + 8N^3$.