SIAM J. OPTIM.
Vol. 22, No. 3, pp. 879–898
© 2012 Society for Industrial and Applied Mathematics

UNCONSTRAINED OPTIMIZATION OF REAL FUNCTIONS IN COMPLEX VARIABLES*

LAURENT SORBER†, MARC VAN BAREL†, AND LIEVEN DE LATHAUWER‡§

Abstract. Nonlinear optimization problems in complex variables are frequently encountered in applied mathematics and engineering applications such as control theory, signal processing, and electrical engineering. Optimization of these problems often requires a first- or second-order approximation of the objective function to generate a new step or descent direction. However, such methods cannot be applied to real functions of complex variables because they are necessarily nonanalytic in their argument, i.e., the Taylor series expansion in their argument alone does not exist. To overcome this problem, the objective function is usually redefined as a function of the real and imaginary parts of its complex argument so that standard optimization methods can be applied. However, this approach may needlessly disguise any inherent structure present in the derivatives of such complex problems. Although little known, it is possible to construct an expansion of the objective function in its original complex variables by noting that functions of complex variables can be analytic in their argument and its complex conjugate as a whole. We use these complex Taylor series expansions to generalize existing optimization algorithms for both general nonlinear optimization problems and nonlinear least squares problems. We then apply these methods to two case studies which demonstrate that complex derivatives can lead to greater insight in the structure of the problem, and that this structure can often be exploited to improve computational complexity and storage cost.

Key words. unconstrained optimization, functions of complex variables, quasi-Newton, BFGS, L-BFGS, nonlinear conjugate gradient, nonlinear least squares, Gauss–Newton, Levenberg–Marquardt, Wirtinger calculus

AMS subject classifications. 90-08, 90C06, 90C53, 90C90, 65K05

DOI. 10.1137/110832124

1. Introduction.

In this article we focus on methods to solve unconstrained nonlinear optimization problems of the form

$$\min_{\boldsymbol{z}\in\mathbb{C}^n} f(\boldsymbol{z},\overline{\boldsymbol{z}}), \tag{1.1}$$

where f is a real, smooth function in n complex variables $\boldsymbol{z}$ and their complex conjugates $\overline{\boldsymbol{z}}$. We will also consider unconstrained nonlinear least squares problems of

*Received by the editors April 27, 2011; accepted for publication (in revised form) March 16, 2012; published electronically July 24, 2012. The scientific responsibility rests with the authors.
http://www.siam.org/journals/siopt/22-3/83212.html

†Department of Computer Science, KU Leuven, Celestijnenlaan 200A, B-3001 Leuven, Belgium (Laurent.Sorber@cs.kuleuven.be, Marc.VanBarel@cs.kuleuven.be). The first author is supported by a doctoral fellowship of the Flanders agency for Innovation by Science and Technology (IWT). The second author's research is supported by (1) the Research Council KU Leuven: (a) project OT/10/038, (b) CoE EF/05/006 Optimization in Engineering (OPTEC), and by (2) the Belgian Network DYSCO (Dynamical Systems, Control, and Optimization), funded by the Interuniversity Attraction Poles Programme, initiated by the Belgian State, Science Policy Office.

‡Group Science, Engineering and Technology, KU Leuven Kulak, E. Sabbelaan 53, B-8500 Kortrijk, Belgium (Lieven.DeLathauwer@kuleuven-kulak.be).

§Department of Electrical Engineering (ESAT), KU Leuven, Kasteelpark Arenberg 10, B-3001 Leuven, Belgium (Lieven.DeLathauwer@esat.kuleuven.be). This author's research is supported by (1) the Research Council KU Leuven: (a) GOA-MaNet, (b) CoE EF/05/006 Optimization in Engineering (OPTEC), (c) CIF1, (d) STRT1/08/023, (2) the Research Foundation Flanders (FWO): (a) project G.0427.10N, (b) Research Communities ICCoS, ANMMM, and MLDM, (3) the Belgian Network DYSCO, and by (4) EU: ERNSI.

the form

$$\min_{\boldsymbol{z}\in\mathbb{C}^n} \frac{1}{2}\|F(\boldsymbol{z},\overline{\boldsymbol{z}})\|^2, \tag{1.2}$$

where F maps n complex variables $\boldsymbol{z}$ and their complex conjugates $\overline{\boldsymbol{z}}$ to m complex residuals $F(\boldsymbol{z},\overline{\boldsymbol{z}})$. Many nonlinear optimization methods use a first- or second-order approximation of the objective function to generate a new step or a descent direction, where the approximation is either updated or recomputed every iteration. On the other hand, the Cauchy–Riemann conditions assert that real-valued functions f in complex variables $\boldsymbol{z}$ are necessarily nonanalytic in $\boldsymbol{z}$. In other words, there exists no Taylor series in $\boldsymbol{z}$ of f at $\boldsymbol{z}_0$ so that the series converges to $f(\boldsymbol{z})$ in a neighborhood of $\boldsymbol{z}_0$. A common workaround is to convert the optimization problem to the real domain by regarding f as a function of the real and imaginary parts of $\boldsymbol{z}$. However, by reformulating an optimization problem that is inherently complex to the real domain, it becomes easy to miss important insights about the structure of the problem which might otherwise be exploited. By making the dependence of the objective function on both $\boldsymbol{z}$ and $\overline{\boldsymbol{z}}$ explicit as we have done in (1.1) and (1.2), we will see that there are several ways to expand these functions into a complex Taylor series. The key to their construction lies in the fact that if a function is analytic in the space spanned by $\mathrm{Re}\{\boldsymbol{z}\}$ and $\mathrm{Im}\{\boldsymbol{z}\}$, it is also analytic in the space spanned by $\boldsymbol{z}$ and $\overline{\boldsymbol{z}}$. Indeed, there is a simple linear transformation from the real to the complex Taylor series. The resulting expansions allow us to generalize existing real optimization methods to the complex domain, and importantly, depend on complex derivatives that are often described by more elegant expressions than their real counterparts.

This paper is organized as follows. In section 2 we first review the notation and give a short overview of Wirtinger calculus, which is the underlying framework for the complex derivatives used in this article, and of complex Taylor series expansions. In section 3 we use these expansions to generalize nonlinear optimization methods such as BFGS, limited memory BFGS, and the nonlinear conjugate gradient algorithm to functions of complex variables. In section 4 we do the same for nonlinear least squares methods such as Gauss–Newton and Levenberg–Marquardt. In section 5 we demonstrate the potential of these generalized optimization methods with two case studies. The first is the canonical polyadic decomposition, which is a tensor decomposition in rank-one terms. The second is the simulation of nonlinear circuits in the frequency domain. We conclude this paper in section 6.

2. Wirtinger calculus and the complex Taylor series expansions.

2.1. Notation and preliminaries. Vectors are denoted by boldface letters and are lower case, e.g., $\boldsymbol{a}$. Matrices are denoted by capital letters, e.g., A. Higher-order tensors are denoted by Euler script letters, e.g., $\mathcal{A}$. The ith entry of a vector $\boldsymbol{a}$ is denoted by a_i, element (i,j) of a matrix A by a_{ij}, and element (i,j,k) of a third-order tensor $\mathcal{A}$ by a_{ijk}. Indices typically range from one to their capital version, e.g., $i = 1,\ldots,I$. A colon is used to indicate all elements of a mode. Thus $\boldsymbol{a}_{:j}$ corresponds to the jth column of a matrix A, which we also denote more compactly as $\boldsymbol{a}_j$. The nth element in a sequence is denoted by a superscript in parentheses, e.g., $A^{(n)}$ denotes the nth matrix in a sequence. The superscripts $\cdot^{\mathrm{T}}$, $\cdot^{\mathrm{H}}$, $\cdot^{-1}$, and $\cdot^{\dagger}$ are used for the transpose, Hermitian conjugate, matrix inverse, and Moore–Penrose pseudoinverse, respectively. The identity matrix of order n is denoted by $\mathbb{I}_n$ and the $m \times n$ all-zero matrix by $0_{m\times n}$. The complex conjugate is denoted by an overline, e.g., $\overline{\boldsymbol{a}}^{\mathrm{T}}$ is equivalent to $\boldsymbol{a}^{\mathrm{H}}$. The two-norm and Frobenius norm are denoted by $\|\cdot\|$ and $\|\cdot\|_F$,

respectively. We use parentheses to denote the concatenation of two or more vectors, e.g., $(\boldsymbol{a}, \boldsymbol{b})$ is equivalent to $[\boldsymbol{a}^{\mathrm{T}} \quad \boldsymbol{b}^{\mathrm{T}}]^{\mathrm{T}}$.

The calculus underlying the complex derivatives in this article goes back to H. Poincaré and was developed principally by W. Wirtinger in the early 20th century. It is often called, especially in the German literature, the Wirtinger calculus [42]. Brandwood first introduced the notion of a complex gradient in the context of optimization [3], though without making the connection with Wirtinger derivatives. Later, Abatzoglou, Mendel, and Harada generalized Brandwood's idea to construct a second-order Taylor series expansion [1]. Their expansion is defined in the function's complex arguments and their complex conjugates separately, resulting in a sum of four second-order terms and no clear definition of a complex Hessian. The complex Taylor series as presented in this article is due to van den Bos [50], who found a compact way of transforming the real gradient and Hessian into their complex counterparts. Kreutz-Delgado went on to point out that there is actually more than one way to define the complex Hessian [26]. We present the two most intuitive definitions, of which one in particular is useful from an optimization point of view.

In order to facilitate the transition from and to the real and complex numbers, we first define the vector spaces $\mathcal{R}$, $\mathcal{C}$, and $\mathcal{C}^*$ as $\{(\boldsymbol{x}, \boldsymbol{y}) \,|\boldsymbol{x} \in \mathbb{R}^n, \boldsymbol{y} \in \mathbb{R}^n\} = \mathbb{R}^{2n}$, $\{(\boldsymbol{z}, \overline{\boldsymbol{z}}) \,|\boldsymbol{z} \in \mathbb{C}^n\} \subset \mathbb{C}^{2n}$, and $\{(\overline{\boldsymbol{z}}, \boldsymbol{z}) \,|\boldsymbol{z} \in \mathbb{C}^n\} \subset \mathbb{C}^{2n}$, respectively. Elements of $\mathbb{C}^n$ have an equivalent representation in any of these spaces. Let $\boldsymbol{z} \in \mathbb{C}^n$; then we define $\overset{\mathcal{R}}{\boldsymbol{z}} \triangleq (\mathrm{Re}\{\boldsymbol{z}\}, \mathrm{Im}\{\boldsymbol{z}\}) \in \mathcal{R}$, $\overset{\mathcal{C}}{\boldsymbol{z}} \triangleq (\boldsymbol{z}, \overline{\boldsymbol{z}}) \in \mathcal{C}$ and $\overset{\mathcal{C}^*}{\boldsymbol{z}} \triangleq (\overline{\boldsymbol{z}}, \boldsymbol{z}) = \overset{\mathcal{C}}{\overline{\boldsymbol{z}}} \in \mathcal{C}$. Furthermore, the linear map

$$J \triangleq \begin{bmatrix} \mathbb{I}_n & i\mathbb{I}_n \\ \mathbb{I}_n & -i\mathbb{I}_n \end{bmatrix} \tag{2.1}$$

is an isomorphism from $\mathcal{R}$ to $\mathcal{C}$ and its inverse is given by $J^{-1} = \frac{1}{2}J^{\mathrm{H}}$. The *swap operator*

$$S \triangleq \begin{bmatrix} 0 & \mathbb{I}_n \\ \mathbb{I}_n & 0 \end{bmatrix} \tag{2.2}$$

is an isomorphism from $\mathcal{C}$ to the *dual space* $\mathcal{C}^*$. Its inverse is given by $S^{-1} = S^{\mathrm{T}} = S$.

DEFINITION 2.1. *Let* $\boldsymbol{z} \in \mathbb{C}^n$ *and let* $\boldsymbol{x} = \mathrm{Re}\{\boldsymbol{z}\}$ *and* $\boldsymbol{y} = \mathrm{Im}\{\boldsymbol{z}\}$. *The* cogradient operator $\frac{\partial}{\partial \boldsymbol{z}}$ *and* conjugate cogradient operator $\frac{\partial}{\partial \overline{\boldsymbol{z}}}$ *are defined as* [3, 26, 42, 50]

$$\frac{\partial}{\partial \boldsymbol{z}} \triangleq \frac{1}{2}\begin{bmatrix} \frac{\partial}{\partial x_1} - \frac{\partial}{\partial y_1}i \\ \vdots \\ \frac{\partial}{\partial x_n} - \frac{\partial}{\partial y_n}i \end{bmatrix}, \tag{2.3a}$$

$$\frac{\partial}{\partial \overline{\boldsymbol{z}}} \triangleq \frac{1}{2}\begin{bmatrix} \frac{\partial}{\partial x_1} + \frac{\partial}{\partial y_1}i \\ \vdots \\ \frac{\partial}{\partial x_n} + \frac{\partial}{\partial y_n}i \end{bmatrix}. \tag{2.3b}$$

The (conjugate) cogradient operator acts as a partial derivative with respect to $\boldsymbol{z}$ ($\overline{\boldsymbol{z}}$), treating $\overline{\boldsymbol{z}}$ ($\boldsymbol{z}$) as a constant. To see this, let $z \in \mathbb{C}$ and let $x = \mathrm{Re}\{z\}$ and $y = \mathrm{Im}\{z\}$ so that $x = \frac{1}{2}(z + \overline{z})$ and $y = \frac{i}{2}(\overline{z} - z)$. Then for a function $f : \mathbb{C} \to \mathbb{C}$ we have that $\frac{\partial f}{\partial z} = \frac{\partial f}{\partial x}\frac{\partial x}{\partial z} + \frac{\partial f}{\partial y}\frac{\partial y}{\partial z}$, which is equal to $\frac{1}{2}(\frac{\partial f}{\partial x} - \frac{\partial f}{\partial y}i)$ if $\frac{\partial \overline{z}}{\partial z}$ is set to zero. Note that the Cauchy–Riemann conditions for f to be analytic in z can be expressed

compactly using the cogradient as $\frac{\partial f}{\partial \overline{z}} = 0$, i.e., f is a function only of z. Analogously, f is analytic in $\overline{z}$ if and only if $\frac{\partial f}{\partial z} = 0$.

Although their definitions often allow the cogradients to be expressed elegantly in terms of $\boldsymbol{z}$ and $\overline{\boldsymbol{z}}$, neither contains enough information by itself to express the change in a function with respect to a change in $\boldsymbol{z}$. This motivates the following definition of a complex gradient operator.

DEFINITION 2.2. *Let $\boldsymbol{z} \in \mathbb{C}^n$. We define the* complex gradient operator $\frac{\partial}{\partial \overset{\mathcal{C}}{\boldsymbol{z}}}$ *as*

$$\frac{\partial}{\partial \overset{\mathcal{C}}{\boldsymbol{z}}} \triangleq \left(\frac{\partial}{\partial \boldsymbol{z}}, \frac{\partial}{\partial \overline{\boldsymbol{z}}} \right). \tag{2.4}$$

The linear map (2.1) also defines a one-to-one correspondence between the real gradient $\frac{\partial}{\partial \overset{\mathcal{R}}{\boldsymbol{z}}}$ and the complex gradient $\frac{\partial}{\partial \overset{\mathcal{C}}{\boldsymbol{z}}}$, namely,

$$\frac{\partial}{\partial \overset{\mathcal{R}}{\boldsymbol{z}}} = J^{\mathrm{T}} \frac{\partial}{\partial \overset{\mathcal{C}}{\boldsymbol{z}}}. \tag{2.5}$$

Similarly, the real Hessian $\frac{\partial^2}{\partial \overset{\mathcal{R}}{\boldsymbol{z}} \partial \overset{\mathcal{R}}{\boldsymbol{z}}^{\mathrm{T}}}$ can be transformed into several complex Hessians, two of which are

$$\frac{\partial^2}{\partial \overset{\mathcal{R}}{\boldsymbol{z}} \partial \overset{\mathcal{R}}{\boldsymbol{z}}^{\mathrm{T}}} = \frac{\partial}{\partial \overset{\mathcal{R}}{\boldsymbol{z}}} \left(\frac{\partial}{\partial \overset{\mathcal{R}}{\boldsymbol{z}}} \right)^{\mathrm{T}} = J^{\mathrm{T}} \frac{\partial}{\partial \overset{\mathcal{C}}{\boldsymbol{z}}} \left(\frac{\partial}{\partial \overset{\mathcal{C}}{\boldsymbol{z}}^{\mathrm{T}}} J \right) = J^{\mathrm{T}} \frac{\partial^2}{\partial \overset{\mathcal{C}}{\boldsymbol{z}} \partial \overset{\mathcal{C}}{\boldsymbol{z}}^{\mathrm{T}}} J, \tag{2.6a}$$

$$\frac{\partial^2}{\partial \overset{\mathcal{R}}{\boldsymbol{z}} \partial \overset{\mathcal{R}}{\boldsymbol{z}}^{\mathrm{T}}} = \frac{\partial}{\partial \overset{\mathcal{R}}{\boldsymbol{z}}} \left(\frac{\partial}{\partial \overset{\mathcal{R}}{\boldsymbol{z}}} \right)^{\mathrm{T}} = J^{\mathrm{H}} \frac{\partial}{\partial \overline{\overset{\mathcal{C}}{\boldsymbol{z}}}} \left(\frac{\partial}{\partial \overset{\mathcal{C}}{\boldsymbol{z}}^{\mathrm{T}}} J \right) = J^{\mathrm{H}} \frac{\partial^2}{\partial \overline{\overset{\mathcal{C}}{\boldsymbol{z}}} \partial \overset{\mathcal{C}}{\boldsymbol{z}}^{\mathrm{T}}} J. \tag{2.6b}$$

2.2. First-order complex Taylor series expansion. Consider the first-order real Taylor series expansion of a function $F : \mathcal{R} \to \mathbb{C}^m$,

$$m_F(\Delta \overset{\mathcal{R}}{\boldsymbol{z}}) = F(\overset{\mathcal{R}}{\boldsymbol{z}}) + \frac{\partial F(\overset{\mathcal{R}}{\boldsymbol{z}})}{\partial \overset{\mathcal{R}}{\boldsymbol{z}}^{\mathrm{T}}} \Delta \overset{\mathcal{R}}{\boldsymbol{z}}. \tag{2.7}$$

Because $\mathcal{R}$ and $\mathcal{C}$ are isomorphic, the function F can also be regarded as a function of $\overset{\mathcal{C}}{\boldsymbol{z}}$. Although it is generally not true that F is analytic in $\boldsymbol{z}$ and $\overline{\boldsymbol{z}}$ independently if F is analytic in $\overset{\mathcal{R}}{\boldsymbol{z}}$, it does hold that F is analytic in $\boldsymbol{z}$ and $\overline{\boldsymbol{z}}$ as a whole. Using (2.1) and (2.5), the first-order complex Taylor series expansion of F can be expressed as

$$m_F(\Delta \overset{\mathcal{C}}{\boldsymbol{z}}) = F(\overset{\mathcal{C}}{\boldsymbol{z}}) + \frac{\partial F(\overset{\mathcal{C}}{\boldsymbol{z}})}{\partial \overset{\mathcal{C}}{\boldsymbol{z}}^{\mathrm{T}}} \Delta \overset{\mathcal{C}}{\boldsymbol{z}}. \tag{2.8}$$

The matrix $\frac{\partial F}{\partial \overset{\mathcal{C}}{\boldsymbol{z}}^{\mathrm{T}}} = \begin{bmatrix} \frac{\partial F}{\partial \boldsymbol{z}^{\mathrm{T}}} & \frac{\partial F}{\partial \overline{\boldsymbol{z}}^{\mathrm{T}}} \end{bmatrix}$ is obtained by applying the transpose of the complex gradient operator (2.5) componentwise to F. The matrices $\frac{\partial F}{\partial \boldsymbol{z}^{\mathrm{T}}}$ and $\frac{\partial F}{\partial \overline{\boldsymbol{z}}^{\mathrm{T}}}$ are called the *Jacobian* and *conjugate Jacobian*, respectively.

2.3. Second-order complex Taylor series expansion. Consider the second-order real Taylor series expansion of a function $f : \mathcal{R} \to \mathbb{C}$,

$$m_f(\Delta \overset{\mathcal{R}}{\boldsymbol{z}}) = f(\overset{\mathcal{R}}{\boldsymbol{z}}) + \Delta \overset{\mathcal{R}}{\boldsymbol{z}}^{\mathrm{T}} \frac{\partial f(\overset{\mathcal{R}}{\boldsymbol{z}})}{\partial \overset{\mathcal{R}}{\boldsymbol{z}}} + \frac{1}{2} \Delta \overset{\mathcal{R}}{\boldsymbol{z}}^{\mathrm{T}} \frac{\partial^2 f(\overset{\mathcal{R}}{\boldsymbol{z}})}{\partial \overset{\mathcal{R}}{\boldsymbol{z}} \partial \overset{\mathcal{R}}{\boldsymbol{z}}^{\mathrm{T}}} \Delta \overset{\mathcal{R}}{\boldsymbol{z}}. \tag{2.9}$$

Because $\mathcal{R}$ and $\mathcal{C}$ are isomorphic, the function f can also be regarded as a function of $\overset{\mathcal{C}}{\boldsymbol{z}}$. Using (2.1), (2.5), and (2.6), the second-order Taylor series expansion of $f(\overset{\mathcal{C}}{\boldsymbol{z}})$

can be expressed in the following two equivalent ways:

$$m_f(\Delta\overset{\mathbb{C}}{\boldsymbol{z}}) = f(\overset{\mathbb{C}}{\boldsymbol{z}}) + \Delta\overset{\mathbb{C}}{\boldsymbol{z}}^{\mathrm{T}}\frac{\partial f(\overset{\mathbb{C}}{\boldsymbol{z}})}{\partial \overset{\mathbb{C}}{\boldsymbol{z}}} + \frac{1}{2}\Delta\overset{\mathbb{C}}{\boldsymbol{z}}^{\mathrm{T}}\frac{\partial^2 f(\overset{\mathbb{C}}{\boldsymbol{z}})}{\partial \overset{\mathbb{C}}{\boldsymbol{z}}\partial \overset{\mathbb{C}}{\boldsymbol{z}}^{\mathrm{T}}}\Delta\overset{\mathbb{C}}{\boldsymbol{z}}, \tag{2.10a}$$

$$m_f(\Delta\overset{\mathbb{C}}{\boldsymbol{z}}) = f(\overset{\mathbb{C}}{\boldsymbol{z}}) + \Delta\overset{\mathbb{C}}{\boldsymbol{z}}^{\mathrm{T}}\frac{\partial f(\overset{\mathbb{C}}{\boldsymbol{z}})}{\partial \overset{\mathbb{C}}{\boldsymbol{z}}} + \frac{1}{2}\Delta\overset{\mathbb{C}}{\boldsymbol{z}}^{\mathrm{H}}\frac{\partial^2 f(\overset{\mathbb{C}}{\boldsymbol{z}})}{\partial \overset{\mathbb{C}}{\overline{\boldsymbol{z}}}\partial \overset{\mathbb{C}}{\boldsymbol{z}}^{\mathrm{T}}}\Delta\overset{\mathbb{C}}{\boldsymbol{z}}. \tag{2.10b}$$

Often $f(\overset{\mathbb{R}}{\boldsymbol{z}})$ will have continuous second-order derivatives. Clairaut's theorem [22] states that the order of differentiation does not matter for functions for which this property holds, and hence that the real Hessian $\frac{\partial^2 f}{\partial \overset{\mathbb{R}}{\boldsymbol{z}}\partial \overset{\mathbb{R}}{\boldsymbol{z}}^{\mathrm{T}}}$ is symmetric. It then follows from (2.6) that $\frac{\partial^2 f}{\partial \overset{\mathbb{C}}{\boldsymbol{z}}\partial \overset{\mathbb{C}}{\boldsymbol{z}}^{\mathrm{T}}}$ is symmetric and $\frac{\partial^2 f}{\partial \overset{\mathbb{C}}{\overline{\boldsymbol{z}}}\partial \overset{\mathbb{C}}{\boldsymbol{z}}^{\mathrm{T}}}$ is Hermitian for real-valued f.

Note that (2.10b) is not differentiable with respect to $\Delta\overset{\mathbb{C}}{\boldsymbol{z}}$ because of the dependency on $\Delta\overset{\mathbb{C}}{\boldsymbol{z}}^{\mathrm{H}}$. In contrast, (2.10a) is differentiable with respect to $\Delta\overset{\mathbb{C}}{\boldsymbol{z}}$. To overcome this apparent difficulty, we note that the dependency on $\Delta\overset{\mathbb{C}}{\boldsymbol{z}}^{\mathrm{H}}$ can be resolved by substituting $\Delta\overset{\mathbb{C}}{\boldsymbol{z}}^{\mathrm{H}} = \Delta\overset{\mathbb{C}}{\overline{\boldsymbol{z}}}^{\mathrm{T}} = \Delta\overset{\mathbb{C}}{\boldsymbol{z}}^{\mathrm{T}}S$. The complex gradient of the second-order Taylor expansions (2.10) can now be computed as

$$\frac{\partial m_f(\Delta\overset{\mathbb{C}}{\boldsymbol{z}})}{\partial \Delta\overset{\mathbb{C}}{\boldsymbol{z}}} = \frac{\partial f(\overset{\mathbb{C}}{\boldsymbol{z}})}{\partial \overset{\mathbb{C}}{\boldsymbol{z}}} + \frac{\partial^2 f(\overset{\mathbb{C}}{\boldsymbol{z}})}{\partial \overset{\mathbb{C}}{\boldsymbol{z}}\partial \overset{\mathbb{C}}{\boldsymbol{z}}^{\mathrm{T}}}\Delta\overset{\mathbb{C}}{\boldsymbol{z}} \tag{2.11a}$$

and

$$\frac{\partial m_f(\Delta\overset{\mathbb{C}}{\boldsymbol{z}})}{\partial \Delta\overset{\mathbb{C}}{\boldsymbol{z}}} = \frac{\partial f(\overset{\mathbb{C}}{\boldsymbol{z}})}{\partial \overset{\mathbb{C}}{\boldsymbol{z}}} + \frac{1}{2}\left(S\frac{\partial^2 f(\overset{\mathbb{C}}{\boldsymbol{z}})}{\partial \overset{\mathbb{C}}{\overline{\boldsymbol{z}}}\partial \overset{\mathbb{C}}{\boldsymbol{z}}^{\mathrm{T}}} + \left(\frac{\partial^2 f(\overset{\mathbb{C}}{\boldsymbol{z}})}{\partial \overset{\mathbb{C}}{\overline{\boldsymbol{z}}}\partial \overset{\mathbb{C}}{\boldsymbol{z}}^{\mathrm{T}}}\right)^{\mathrm{T}} S\right)\Delta\overset{\mathbb{C}}{\boldsymbol{z}}, \tag{2.11b}$$

respectively.

Let us now assume f maps to the real numbers. This assumption has certain consequences for the structure of the gradient and Hessian in the Taylor series expansions (2.10). First, because of the identity $\frac{\partial \overline{f}}{\partial \boldsymbol{z}} = \overline{\frac{\partial f}{\partial \overline{\boldsymbol{z}}}}$ [31, 42], it follows that $\frac{\partial f}{\partial \boldsymbol{z}} = \overline{\frac{\partial f}{\partial \overline{\boldsymbol{z}}}}$ for real-valued f. Second, it can also be shown that $\frac{\partial^2 f}{\partial \boldsymbol{z}\partial \overline{\boldsymbol{z}}^{\mathrm{T}}} = \frac{\partial^2 f}{\partial \overline{\boldsymbol{z}}\partial \boldsymbol{z}^{\mathrm{T}}}$ and $\frac{\partial^2 f}{\partial \boldsymbol{z}\partial \boldsymbol{z}^{\mathrm{T}}} = \overline{\frac{\partial^2 f}{\partial \overline{\boldsymbol{z}}\partial \overline{\boldsymbol{z}}^{\mathrm{T}}}}$. Using these properties, we can simplify expression (2.11b) by premultiplying with S, so that the model's conjugate complex gradient is given by

$$S\frac{\partial m_f(\Delta\overset{\mathbb{C}}{\boldsymbol{z}})}{\partial \Delta\overset{\mathbb{C}}{\boldsymbol{z}}} = S\frac{\partial f(\overset{\mathbb{C}}{\boldsymbol{z}})}{\partial \overset{\mathbb{C}}{\boldsymbol{z}}} + \frac{1}{2}\left(SS\frac{\partial^2 f(\overset{\mathbb{C}}{\boldsymbol{z}})}{\partial \overset{\mathbb{C}}{\overline{\boldsymbol{z}}}\partial \overset{\mathbb{C}}{\boldsymbol{z}}^{\mathrm{T}}} + S\left(\frac{\partial^2 f(\overset{\mathbb{C}}{\boldsymbol{z}})}{\partial \overset{\mathbb{C}}{\overline{\boldsymbol{z}}}\partial \overset{\mathbb{C}}{\boldsymbol{z}}^{\mathrm{T}}}\right)^{\mathrm{T}} S\right)\Delta\overset{\mathbb{C}}{\boldsymbol{z}},$$

$$\overline{\frac{\partial m_f(\Delta\overset{\mathbb{C}}{\boldsymbol{z}})}{\partial \Delta\overset{\mathbb{C}}{\boldsymbol{z}}}} = \overline{\frac{\partial f(\overset{\mathbb{C}}{\boldsymbol{z}})}{\partial \overset{\mathbb{C}}{\boldsymbol{z}}}} + \frac{\partial^2 f(\overset{\mathbb{C}}{\boldsymbol{z}})}{\partial \overset{\mathbb{C}}{\overline{\boldsymbol{z}}}\partial \overset{\mathbb{C}}{\boldsymbol{z}}^{\mathrm{T}}}\Delta\overset{\mathbb{C}}{\boldsymbol{z}}. \tag{2.12}$$

3. Nonlinear optimization problems $\min_{\boldsymbol{z}} f(\boldsymbol{z}, \overline{\boldsymbol{z}})$. First we consider nonlinear optimization problems of the form (1.1). The space spanned by $\boldsymbol{z}$ and $\overline{\boldsymbol{z}}$ is equal to $\mathcal{C}$, which is equivalent to $\mathcal{R}$ under the linear map (2.1). It is then understood that the function f can also be regarded as a function that maps $\mathcal{C}$ or $\mathcal{R}$ to $\mathbb{R}$, so that both the second-order model (2.9) in $\overset{\mathbb{R}}{\boldsymbol{z}}$ and the two second-order models (2.10) in $\overset{\mathbb{C}}{\boldsymbol{z}}$ are applicable.

3.1. The generalized BFGS and L-BFGS methods. In the generalized BFGS method, we use the quadratic model

$$m_k(\overset{\mathbb{C}}{\boldsymbol{p}}) = f(\overset{\mathbb{C}}{\boldsymbol{z}}_k) + \overset{\mathbb{C}}{\boldsymbol{p}}^{\mathrm{T}}\frac{\partial f(\overset{\mathbb{C}}{\boldsymbol{z}}_k)}{\partial \overset{\mathbb{C}}{\boldsymbol{z}}} + \frac{1}{2}\overset{\mathbb{C}}{\boldsymbol{p}}^{\mathrm{H}}B_k\overset{\mathbb{C}}{\boldsymbol{p}} \tag{3.1}$$

of the objective function f at the current iterate $\overset{c}{\boldsymbol{z}}_k$, where B_k is a Hermitian positive definite matrix that is updated every iteration. Since f is real-valued, the minimizer of this convex quadratic model can be obtained by setting the conjugate complex gradient (2.12) equal to zero, and is given by

$$\overset{c}{\boldsymbol{p}}_k = -B_k^{-1} \overline{\frac{\partial f(\overset{c}{\boldsymbol{z}}_k)}{\partial \overset{c}{\boldsymbol{z}}}}. \tag{3.2}$$

In a line search framework, the next iterate is $\boldsymbol{z}_{k+1} = \boldsymbol{z}_k + \alpha_k \boldsymbol{p}_k$, where the real step length α_k is usually chosen to satisfy the (strong) Wolfe conditions [33, 51]. A reasonable requirement for the updated Hessian B_{k+1} is that the gradient of the model m_{k+1} matches the gradient of the objective function f in the last two iterates $\overset{c}{\boldsymbol{z}}_k$ and $\overset{c}{\boldsymbol{z}}_{k+1}$. The second condition is satisfied automatically by (3.1). The first condition can be written as

$$\overline{\frac{\partial m_{k+1}(-\alpha_k \overset{c}{\boldsymbol{p}}_k)}{\partial \overset{c}{\boldsymbol{p}}}} = \overline{\frac{\partial f(\overset{c}{\boldsymbol{z}}_{k+1})}{\partial \overset{c}{\boldsymbol{z}}}} - \alpha_k B_{k+1} \overset{c}{\boldsymbol{p}}_k = \overline{\frac{\partial f(\overset{c}{\boldsymbol{z}}_k)}{\partial \overset{c}{\boldsymbol{z}}}}.$$

Define $\overset{c}{\boldsymbol{y}}_k \triangleq \overline{\frac{\partial f(\overset{c}{\boldsymbol{z}}_{k+1})}{\partial \overset{c}{\boldsymbol{z}}}} - \overline{\frac{\partial f(\overset{c}{\boldsymbol{z}}_k)}{\partial \overset{c}{\boldsymbol{z}}}}$ and $\overset{c}{\boldsymbol{s}}_k \triangleq \overset{c}{\boldsymbol{z}}_{k+1} - \overset{c}{\boldsymbol{z}}_k$, then by rearranging we find that B_{k+1} should satisfy the *secant equation*

$$B_{k+1} \overset{c}{\boldsymbol{s}}_k = \overset{c}{\boldsymbol{y}}_k. \tag{3.3}$$

The BFGS update [5, 11, 15, 45] is chosen so that the inverse of the updated Hessian B_{k+1}^{-1} is, among all symmetric positive definite matrices satisfying the secant equation (3.3), in some sense closest to the current inverse Hessian approximation B_k^{-1}. In Lemma 3.2 and Theorem 3.3, we generalize Schnabel and Dennis' derivation of the BFGS update [44] to Hermitian Hessians. The proof requires the Broyden update [10], which is generalized to the complex space in Theorem 3.1.

THEOREM 3.1. *Let $B \in \mathbb{C}^{n \times n}$, $\boldsymbol{y}, \boldsymbol{s} \in \mathbb{C}^n$, and $\boldsymbol{s}$ be nonzero. Then the unique minimizer of*

$$\begin{aligned} &\min_{\hat{B} \in \mathbb{C}^{n \times n}} \|B - \hat{B}\|_F \\ &\text{s.t. } \hat{B}\boldsymbol{s} = \boldsymbol{y} \end{aligned} \tag{3.4}$$

is given by the Broyden update

$$B^* = B + \frac{(\boldsymbol{y} - B\boldsymbol{s})\boldsymbol{s}^{\mathrm{H}}}{\boldsymbol{s}^{\mathrm{H}}\boldsymbol{s}}. \tag{3.5}$$

Proof. To show that B^* is a solution to (3.4), note that $B^*\boldsymbol{s} = \boldsymbol{y}$ and that

$$\|B - B^*\|_F = \left\| (B - \hat{B}) \frac{\boldsymbol{s}\boldsymbol{s}^{\mathrm{H}}}{\boldsymbol{s}^{\mathrm{H}}\boldsymbol{s}} \right\|_F \leq \|B - \hat{B}\|_F.$$

That B^* is the unique solution follows from the fact that the mapping $f : \mathbb{C}^{n \times n} \to \mathbb{R}$ defined by $f(\hat{B}) = \|B - \hat{B}\|_F$ is strictly convex in $\mathbb{C}^{n \times n}$ and that the set of $\hat{B} \in \mathbb{C}^{n \times n}$ such that $\hat{B}\boldsymbol{s} = \boldsymbol{y}$ is convex. ∎

LEMMA 3.2. *Let $\boldsymbol{y}, \boldsymbol{s} \in \mathbb{C}^n$, $\boldsymbol{s}$ nonzero and let $Q(\boldsymbol{y}, \boldsymbol{s}) \triangleq \{B \in \mathbb{C}^{n \times n} | B\boldsymbol{s} = \boldsymbol{y}\}$. Then the* generalized quotient *$Q(\boldsymbol{y}, \boldsymbol{s})$ contains a Hermitian positive definite matrix*

if and only if for some nonzero $\boldsymbol{v} \in \mathbb{C}^n$ *and nonsingular* $J \in \mathbb{C}^{n\times n}$, $\boldsymbol{y} = J\boldsymbol{v}$, *and* $\boldsymbol{v} = J^{\mathrm{H}}\boldsymbol{s}$.

Proof. If $\boldsymbol{v}$ and J exist, then $\boldsymbol{y} = J\boldsymbol{v} = JJ^{\mathrm{H}}\boldsymbol{s}$ and JJ^{H} is a Hermitian positive definite matrix in $Q(\boldsymbol{y}, \boldsymbol{s})$. Now suppose B is a Hermitian positive definite matrix which satisfies $B\boldsymbol{s} = \boldsymbol{y}$. Let $B = LL^{\mathrm{H}}$ be the Cholesky factorization of B and set $J = L$ and $\boldsymbol{v} = L^{\mathrm{H}}\boldsymbol{s}$ to complete the proof. ∎

THEOREM 3.3. *Let* $L \in \mathbb{C}^{n\times n}$ *be nonsingular,* $B = LL^{\mathrm{H}}$, $\boldsymbol{y}, \boldsymbol{s} \in \mathbb{C}^n$, *and* $\boldsymbol{s}$ *be nonzero. There is a Hermitian positive definite matrix* $\hat{B} \in Q(\boldsymbol{y}, \boldsymbol{s})$ *if and only if* $\boldsymbol{y}^{\mathrm{H}}\boldsymbol{s} > 0$. *If there is such a matrix, then the* generalized BFGS update $B^* \triangleq \hat{L}\hat{L}^{\mathrm{H}}$ *is one such, where*

$$\hat{L} = \left(\mathbb{I}_n \pm \sqrt{\frac{\boldsymbol{y}^{\mathrm{H}}\boldsymbol{s}}{\boldsymbol{s}^{\mathrm{H}}B\boldsymbol{s}}}\frac{\boldsymbol{y}\boldsymbol{s}^{\mathrm{H}}}{\boldsymbol{y}^{\mathrm{H}}\boldsymbol{s}} - \frac{B\boldsymbol{s}\boldsymbol{s}^{\mathrm{H}}}{\boldsymbol{s}^{\mathrm{H}}B\boldsymbol{s}}\right) L, \tag{3.6}$$

so that

$$B^* = B - \frac{B\boldsymbol{s}\boldsymbol{s}^{\mathrm{H}}B}{\boldsymbol{s}^{\mathrm{H}}B\boldsymbol{s}} + \frac{\boldsymbol{y}\boldsymbol{y}^{\mathrm{H}}}{\boldsymbol{y}^{\mathrm{H}}\boldsymbol{s}} \tag{3.7}$$

and

$$B^{*-1} = \left(\mathbb{I}_n - \frac{\boldsymbol{s}\boldsymbol{y}^{\mathrm{H}}}{\boldsymbol{y}^{\mathrm{H}}\boldsymbol{s}}\right) B^{-1} \left(\mathbb{I}_n - \frac{\boldsymbol{y}\boldsymbol{s}^{\mathrm{H}}}{\boldsymbol{y}^{\mathrm{H}}\boldsymbol{s}}\right) + \frac{\boldsymbol{s}\boldsymbol{s}^{\mathrm{H}}}{\boldsymbol{y}^{\mathrm{H}}\boldsymbol{s}}. \tag{3.8}$$

Proof. Let $\hat{B}$ be a Hermitian positive definite matrix in $Q(\boldsymbol{y}, \boldsymbol{s})$; then $\boldsymbol{s}^{\mathrm{H}}\boldsymbol{y} = \boldsymbol{s}^{\mathrm{H}}\hat{B}\boldsymbol{s} > 0$ is a necessary condition for the update to exist. The nearest matrix $\hat{L}$ in Frobenius norm to L that satisfies $\hat{L}\boldsymbol{v} = \boldsymbol{y}$ is given by the Broyden update (3.5),

$$\hat{L} = L + \frac{(\boldsymbol{y} - L\boldsymbol{v})\boldsymbol{v}^{\mathrm{H}}}{\boldsymbol{v}^{\mathrm{H}}\boldsymbol{v}}.$$

If we can find a $\boldsymbol{v} \in \mathbb{C}^n$ such that $\boldsymbol{v} = \hat{L}^{\mathrm{H}}\boldsymbol{s}$, then by Lemma 3.2 we know that $\hat{L}\hat{L}^{\mathrm{H}}$ is in $Q(\boldsymbol{y}, \boldsymbol{s})$. The condition

$$\boldsymbol{v} = \hat{L}^{\mathrm{H}}\boldsymbol{s} = L^{\mathrm{H}}\boldsymbol{s} + \frac{(\boldsymbol{y}^{\mathrm{H}}\boldsymbol{s} - \boldsymbol{v}^{\mathrm{H}}L^{\mathrm{H}}\boldsymbol{s})}{\boldsymbol{v}^{\mathrm{H}}\boldsymbol{v}}\boldsymbol{v}$$

implies that $\boldsymbol{v} = \alpha L^{\mathrm{H}}\boldsymbol{s}$ for some scalar α. Plugging this back into (x), we find

$$\alpha = 1 + \frac{\alpha\boldsymbol{y}^{\mathrm{H}}\boldsymbol{s} - |\alpha|^2\boldsymbol{s}^{\mathrm{H}}B\boldsymbol{s}}{|\alpha|^2\boldsymbol{s}^{\mathrm{H}}B\boldsymbol{s}}$$

or

$$|\alpha|^2 = \frac{\boldsymbol{y}^{\mathrm{H}}\boldsymbol{s}}{\boldsymbol{s}^{\mathrm{H}}B\boldsymbol{s}},$$

which shows that $\boldsymbol{y}^{\mathrm{H}}\boldsymbol{s} > 0$ is a sufficient condition for the update to exist. Filling $\boldsymbol{v} = \alpha L^{\mathrm{H}}\boldsymbol{s}$ back in the Broyden update results in (3.6). It is then a matter of verifying (3.7)–(3.8) by forming the product $\hat{L}\hat{L}^{\mathrm{H}}$ and applying the Sherman–Morrison formula [46, 47], respectively. ∎

Theorem 3.3 holds for any $\boldsymbol{s}$ and $\boldsymbol{y}$ in $\mathbb{C}^{2n}$ if all dimensions are scaled appropriately, and hence also for the subset $\mathcal{C} \subset \mathbb{C}^{2n}$. At every step of the generalized BFGS method, the inverse Hessian can then be updated by (3.8), so that

$$B_{k+1}^{-1} = \left(\mathbb{I}_{2n} - \frac{\overset{\mathcal{C}}{\boldsymbol{s}}_k \overset{\mathcal{C}}{\boldsymbol{y}}_k^{\mathrm{H}}}{\overset{\mathcal{C}}{\boldsymbol{y}}_k^{\mathrm{H}} \overset{\mathcal{C}}{\boldsymbol{s}}_k}\right) B_k^{-1} \left(\mathbb{I}_{2n} - \frac{\overset{\mathcal{C}}{\boldsymbol{y}}_k \overset{\mathcal{C}}{\boldsymbol{s}}_k^{\mathrm{H}}}{\overset{\mathcal{C}}{\boldsymbol{y}}_k^{\mathrm{H}} \overset{\mathcal{C}}{\boldsymbol{s}}_k}\right) + \frac{\overset{\mathcal{C}}{\boldsymbol{s}}_k \overset{\mathcal{C}}{\boldsymbol{s}}_k^{\mathrm{H}}}{\overset{\mathcal{C}}{\boldsymbol{y}}_k^{\mathrm{H}} \overset{\mathcal{C}}{\boldsymbol{s}}_k}.$$

If the number of variables n is large, the cost of storing and manipulating the inverse Hessian B_{k+1}^{-1} can become prohibitive. The *limited memory* BFGS (L-BFGS) method [13, 27, 32] circumvents this problem by storing the inverse Hessian implicitly as a set of m vector pairs $\{\overset{\mathcal{C}}{\boldsymbol{s}}_i, \overset{\mathcal{C}}{\boldsymbol{y}}_i\}$, $i = k-m, \ldots, k-1$. In fact, it suffices to store $\{\boldsymbol{s}_i, \boldsymbol{y}_i\}$, since the second half of any vector in $\mathcal{C}$ is just the complex conjugate of its first half.

Suppose we have a real function of complex variables $f : \mathcal{C} \to \mathbb{R}$ that we are interested in minimizing for $\boldsymbol{z} \in \mathbb{C}^n$ as well as for $\boldsymbol{x} \in \mathbb{R}^n$. The quasi-Newton step (3.2) depends on the complex gradient. Because the conjugate cogradient is just the complex conjugate of the cogradient for real-valued f, we need only compute one of the cogradients. In this case, we choose the conjugate cogradient $\frac{\partial f}{\partial \overline{\boldsymbol{z}}}$ because it coincides with the steepest descent direction. For optimization of $\boldsymbol{x} \in \mathbb{R}^n$, we need the real gradient $\frac{\partial f(\boldsymbol{x}_k)}{\partial \boldsymbol{x}}$, which can also be expressed as $\frac{\partial f(\boldsymbol{x}_k)}{\partial \boldsymbol{z}} + \frac{\partial f(\boldsymbol{x}_k)}{\partial \overline{\boldsymbol{z}}} = 2\frac{\partial f(\boldsymbol{x}_k)}{\partial \overline{\boldsymbol{z}}}$. Therefore, by constructing complex optimization algorithms in a way that only requires evaluating objective function values $f(\overset{\mathcal{C}}{\boldsymbol{z}}_k)$ and scaled conjugate cogradients $\boldsymbol{g}(\overset{\mathcal{C}}{\boldsymbol{z}}_k) \triangleq 2\frac{\partial f(\overset{\mathcal{C}}{\boldsymbol{z}}_k)}{\partial \overline{\boldsymbol{z}}}$, the function can be minimized over $\mathbb{C}^n$ as well as $\mathbb{R}^n$ without a separate implementation of the real gradient $\frac{\partial f(\boldsymbol{x}_k)}{\partial \boldsymbol{x}}$.

In Algorithm 3.1 a generalized limited memory BFGS two-loop recursion for computing the quasi-Newton step (3.2) is presented. Its computational and storage cost are equal to that of the limited memory BFGS method applied in the real domain. However, the generalized method requires the objective function's gradient in a form that may be more intuitive to the user, and can furthermore minimize the same objective function over $\mathbb{R}^n$ without requiring the real gradient to be treated as a separate case.

The quasi-Newton step $\boldsymbol{p}_k^*$ computed by Algorithm 3.1 can be used in either a line search or trust-region framework. In the former, the next iterate is $\boldsymbol{z}_{k+1} = \boldsymbol{z}_k + \alpha_k \boldsymbol{p}_k^*$. The step length α_k is usually chosen to loosely minimize the one-dimensional optimization problem $\min_{\alpha_k} f(\overset{\mathcal{C}}{\boldsymbol{z}}_k + \alpha_k \overset{\mathcal{C}}{\boldsymbol{p}}_k)$ so that the *Wolfe conditions* [33, 51]

$$f(\overset{\mathcal{C}}{\boldsymbol{z}}_k + \alpha_k \overset{\mathcal{C}}{\boldsymbol{p}}_k) \leq f(\overset{\mathcal{C}}{\boldsymbol{z}}_k) + c_1 \alpha_k \overset{\mathcal{C}}{\boldsymbol{p}}_k^{\mathrm{T}} \frac{\partial f(\overset{\mathcal{C}}{\boldsymbol{z}}_k)}{\partial \overset{\mathcal{C}}{\boldsymbol{z}}} \tag{3.9a}$$

and

$$\overset{\mathcal{C}}{\boldsymbol{p}}_k^{\mathrm{T}} \frac{\partial f(\overset{\mathcal{C}}{\boldsymbol{z}}_k + \alpha_k \overset{\mathcal{C}}{\boldsymbol{p}}_k)}{\partial \overset{\mathcal{C}}{\boldsymbol{z}}} \geq c_2 \overset{\mathcal{C}}{\boldsymbol{p}}_k^{\mathrm{T}} \frac{\partial f(\overset{\mathcal{C}}{\boldsymbol{z}}_k)}{\partial \overset{\mathcal{C}}{\boldsymbol{z}}}, \tag{3.9b}$$

where $0 < c_1 < c_2 < 1$ are satisfied. Inequalities (3.9a) and (3.9b) are known as the *sufficient decrease* and *curvature condition*, respectively. The former ensures the objective function is sufficiently smaller at the next iterate, and the latter ensures convergence of the gradient to zero. Furthermore, the curvature condition is a sufficient condition for the BFGS update to exist since

$$\overset{\mathcal{C}}{\boldsymbol{s}}_k^{\mathrm{H}} \overset{\mathcal{C}}{\boldsymbol{y}}_k = \alpha_k \overset{\mathcal{C}}{\boldsymbol{p}}_k^{\mathrm{H}} \left(\overline{\frac{\partial f(\overset{\mathcal{C}}{\boldsymbol{z}}_k + \alpha_k \overset{\mathcal{C}}{\boldsymbol{p}}_k)}{\partial \overset{\mathcal{C}}{\boldsymbol{z}}}} - \overline{\frac{\partial f(\overset{\mathcal{C}}{\boldsymbol{z}}_k)}{\partial \overset{\mathcal{C}}{\boldsymbol{z}}}}\right) \geq (c_2 - 1)\alpha_k \overset{\mathcal{C}}{\boldsymbol{p}}_k^{\mathrm{T}} \frac{\partial f(\overset{\mathcal{C}}{\boldsymbol{z}}_k)}{\partial \overset{\mathcal{C}}{\boldsymbol{z}}} > 0.$$

Input: $\boldsymbol{g}_k \triangleq \boldsymbol{g}(\overset{\mathcal{C}}{\boldsymbol{z}}_k) = 2\overline{\frac{\partial f(\overset{\mathcal{C}}{\boldsymbol{z}}_k)}{\partial \boldsymbol{z}}} = 2\frac{\partial f(\overset{\mathcal{C}}{\boldsymbol{z}}_k)}{\partial \overline{\boldsymbol{z}}}$,
$\boldsymbol{s}_i \triangleq \boldsymbol{z}_{i+1} - \boldsymbol{z}_i$,
$\boldsymbol{y}_i \triangleq \boldsymbol{g}_{i+1} - \boldsymbol{g}_i$ and
$\rho_i \triangleq \mathrm{Re}\{\boldsymbol{y}_i^{\mathrm{H}}\boldsymbol{s}_i\}^{-1}$ for $i = k-m, \ldots, k-1$

Output: $\boldsymbol{p}$ where $\overset{\mathcal{C}}{\boldsymbol{p}} = -B_k^{-1}\overline{\frac{\partial f(\overset{\mathcal{C}}{\boldsymbol{z}}_k)}{\partial \overset{\mathcal{C}}{\boldsymbol{z}}}}$

$\boldsymbol{p} \leftarrow -\boldsymbol{g}_k$
for $i = k-1, k-2, \ldots, k-m$ **do**
  $\alpha_i \leftarrow \rho_i \mathrm{Re}\{\boldsymbol{s}_i^{\mathrm{H}}\boldsymbol{p}\}$
  $\boldsymbol{p} \leftarrow \boldsymbol{p} - \alpha_i \boldsymbol{y}_i$
end
$\boldsymbol{p} \leftarrow \frac{1}{2}B_{k-m}^{-1}\boldsymbol{p}$ (e.g., $\frac{1}{2}B_{k-m}^{-1} = \frac{\mathrm{Re}\{\boldsymbol{y}_{k-1}^{\mathrm{H}}\boldsymbol{s}_{k-1}\}}{\boldsymbol{y}_{k-1}^{\mathrm{H}}\boldsymbol{y}_{k-1}}\mathbb{I}_n$ [13])
for $i = k-m, k-m+1, \ldots, k-1$ **do**
  $\beta \leftarrow \rho_i \mathrm{Re}\{\boldsymbol{y}_i^{\mathrm{H}}\boldsymbol{p}\}$
  $\boldsymbol{p} \leftarrow \boldsymbol{p} + (\alpha_i - \beta)\boldsymbol{s}_i$
end

ALGORITHM 3.1. *Generalized L-BFGS two-loop recursion.*

A step length may satisfy the Wolfe conditions without being particularly close to a minimizer of $f(\overset{\mathcal{C}}{\boldsymbol{z}}_k + \alpha_k \overset{\mathcal{C}}{\boldsymbol{p}}_k)$. In the strong Wolfe conditions, the curvature condition is replaced by

$$\left| \overset{\mathcal{C}}{\boldsymbol{p}}_k^{\mathrm{T}} \frac{\partial f(\overset{\mathcal{C}}{\boldsymbol{z}}_k + \alpha_k \overset{\mathcal{C}}{\boldsymbol{p}}_k)}{\partial \overset{\mathcal{C}}{\boldsymbol{z}}} \right| \leq c_2 \left| \overset{\mathcal{C}}{\boldsymbol{p}}_k^{\mathrm{T}} \frac{\partial f(\overset{\mathcal{C}}{\boldsymbol{z}}_k)}{\partial \overset{\mathcal{C}}{\boldsymbol{z}}} \right|$$

so that points far from stationary points are excluded. Line search algorithms are an integral part of quasi-Newton methods, but can be difficult to implement. There are several good software implementations available in the public domain, such as Moré and Thuente [28] and Hager and Zhang [16], which can be generalized to functions in complex variables with relatively little effort. Like Algorithm 3.1, their implementation can be organized such that all computations are in $\mathbb{C}^n$ instead of $\mathcal{C}$. For instance, the Wolfe conditions (3.9) are equivalent to

$$f(\overset{\mathcal{C}}{\boldsymbol{z}}_k + \alpha_k \overset{\mathcal{C}}{\boldsymbol{p}}_k) \leq f(\overset{\mathcal{C}}{\boldsymbol{z}}_k) + c_1 \alpha_k \mathrm{Re}\{\boldsymbol{p}_k^{\mathrm{H}}\boldsymbol{g}_k\} \tag{3.10a}$$

and

$$\mathrm{Re}\left\{\boldsymbol{p}_k^{\mathrm{H}}\boldsymbol{g}(\overset{\mathcal{C}}{\boldsymbol{z}}_k + \alpha_k \overset{\mathcal{C}}{\boldsymbol{p}}_k)\right\} \geq c_2 \mathrm{Re}\{\boldsymbol{p}_k^{\mathrm{H}}\boldsymbol{g}_k\}. \tag{3.10b}$$

In a trust-region framework, a region around the current iterate $\boldsymbol{z}_k$ is defined in which the model m_k is trusted to be an adequate representation of the objective function. The next iterate $\boldsymbol{z}_{k+1}$ is then chosen to be the approximate minimizer of the model in this region. In effect, the direction and length of the step is chosen simultaneously. The trust-region radius Δ_k is updated every iteration based on the trustworthiness ρ_k of the model, which is defined as the ratio of the actual reduction $f(\overset{\mathcal{C}}{\boldsymbol{z}}_k) - f(\overset{\mathcal{C}}{\boldsymbol{z}}_k + \overset{\mathcal{C}}{\boldsymbol{p}}_k)$ and the predicted reduction $m_k(\boldsymbol{0}) - m_k(\overset{\mathcal{C}}{\boldsymbol{p}}_k)$.

There exist several strategies to approximately solve the trust-region subproblem

$$\begin{aligned} &\min_{\boldsymbol{p}\in\mathbb{C}^n} m_k(\overset{\mathcal{C}}{\boldsymbol{p}}) \\ &\text{s.t. } \|\boldsymbol{p}\| \leq \Delta_k. \end{aligned} \tag{3.11}$$

The quasi-Newton step $\boldsymbol{p}_k^*$ minimizes (3.11) when $\|\boldsymbol{p}_k^*\| \leq \Delta_k$. If the trust-region radius is small compared to the quasi-Newton step, the quadratic term in m_k has little effect and the solution to the trust-region subproblem can be approximated by $\boldsymbol{p}^* = -\Delta_k \frac{\boldsymbol{g}_k}{\|\boldsymbol{g}_k\|}$. The dogleg method [36, 37], double-dogleg method [9], and two-dimensional subspace minimization [6] all attempt to approximately minimize (3.11) by restricting $\boldsymbol{p}$ to (a subset of) the two-dimensional subspace spanned by the steepest descent direction $-\boldsymbol{g}_k$ and the quasi-Newton step $\boldsymbol{p}_k^*$ when the model Hessian B_k is positive definite. Two-dimensional subspace minimization can also be adapted for indefinite B_k, though in that case it requires an estimate of the most negative eigenvalue of this matrix. For a comprehensive treatment of trust-region strategies, see [8].

3.2. The generalized nonlinear conjugate gradient method. In the nonlinear conjugate gradient method, search directions $\overset{\mathcal{C}}{\boldsymbol{p}}_k$ are generated by the recurrence relation

$$\overset{\mathcal{C}}{\boldsymbol{p}}_k = -\overset{\mathcal{C}}{\boldsymbol{g}}_k + \beta_k \overset{\mathcal{C}}{\boldsymbol{p}}_{k-1}, \tag{3.12}$$

where $\boldsymbol{g}_k$ is now defined[1] as $\frac{\partial f(\overset{\mathcal{C}}{\boldsymbol{z}}_k)}{\partial \overline{\boldsymbol{z}}}$, $\boldsymbol{p}_0$ is initialized as $-\boldsymbol{g}_0$, and β_k is the conjugate gradient update parameter. Different choices for the scalar β_k correspond to different conjugate gradient methods. The generalized Hestenes–Stiefel [19] update parameter β_k^{HS} can be derived from a special case of the L-BFGS search direction, where $m = 1$, $B_{k-1}^{-1} = \mathbb{I}_{2n}$, and an exact line search, for which $\boldsymbol{g}_k^{\mathrm{H}} \boldsymbol{p}_{k-1} = 0$ for all k, is assumed. We then obtain $\overset{\mathcal{C}}{\boldsymbol{p}}_k = -B_k^{-1} \overset{\mathcal{C}}{\boldsymbol{g}}_k = -\overset{\mathcal{C}}{\boldsymbol{g}}_k + \beta_k^{\mathrm{HS}} \overset{\mathcal{C}}{\boldsymbol{p}}_{k-1}$, where

$$\beta_k^{\mathrm{HS}} = \frac{\mathrm{Re}\{(\boldsymbol{g}_k - \boldsymbol{g}_{k-1})^{\mathrm{H}} \boldsymbol{g}_k\}}{\mathrm{Re}\{(\boldsymbol{g}_k - \boldsymbol{g}_{k-1})^{\mathrm{H}} \boldsymbol{p}_{k-1}\}}. \tag{3.13a}$$

When $\boldsymbol{g}_k^{\mathrm{H}} \boldsymbol{p}_{k-1} = 0$, (3.13a) reduces to the Polak–Ribière update parameter [35]

$$\beta_k^{\mathrm{PR}} = \frac{\mathrm{Re}\{(\boldsymbol{g}_k - \boldsymbol{g}_{k-1})^{\mathrm{H}} \boldsymbol{g}_k\}}{\boldsymbol{g}_{k-1}^{\mathrm{H}} \boldsymbol{g}_{k-1}}. \tag{3.13b}$$

Further, if f is quadratic, $\boldsymbol{g}_k^{\mathrm{H}} \boldsymbol{g}_{k-1} = 0$, and we find the Fletcher–Reeves update parameter [12]

$$\beta_k^{\mathrm{FR}} = \frac{\boldsymbol{g}_k^{\mathrm{H}} \boldsymbol{g}_k}{\boldsymbol{g}_{k-1}^{\mathrm{H}} \boldsymbol{g}_{k-1}}. \tag{3.13c}$$

[1] In section 3.1 it was argued that $\boldsymbol{g}_k = 2\frac{\partial f(\overset{\mathcal{C}}{\boldsymbol{z}}_k)}{\partial \overline{\boldsymbol{z}}}$ is a more practical choice for a computer implementation. Using the latter definition throughout this section is also possible, although the generated steps would be twice as long since it is then implicitly assumed that $B_{k-1}^{-1} = 2\mathbb{I}_{2n}$. One way to take this extra scaling factor into account is by scaling the initial line search step length appropriately. However, as it is inherent to the conjugate gradient method that the search directions it generates are often poorly scaled, the extra factor two can safely be ignored depending on the strategy chosen for the initial step length [33].

Powell [38] showed that the Fletcher–Reeves method is susceptible to jamming. That is, the algorithm could take many short steps without making significant progress to the minimum. The Hestenes–Stiefel and Polak–Ribière methods, which share the common numerator $\mathrm{Re}\{(\boldsymbol{g}_k - \boldsymbol{g}_{k-1})^{\mathrm{H}}\boldsymbol{g}_k\}$, possess a built-in restart feature that addresses the jamming problem [17]. When the step $\boldsymbol{z}_k - \boldsymbol{z}_{k-1}$ is small, the factor $\boldsymbol{g}_k - \boldsymbol{g}_{k-1}$ tends to zero. Hence, β_k becomes small and the new search direction $\boldsymbol{p}_k$ is essentially the steepest descent direction $-\boldsymbol{g}_k$.

With an exact line search, the above conjugate gradient methods are all globally convergent. Gilbert and Nocedal [14] proved that the modified Polak–Ribière method [39], where $\beta_k^{\mathrm{PR}+} = \max\{\beta_k^{\mathrm{PR}}, 0\}$, is globally convergent even when an inexact line search satisfying the Wolfe conditions is used. Similarly, it can also be shown [17] that the modified Hestenes–Stiefel method, where $\beta_k^{\mathrm{HS}+} = \max\{\beta_k^{\mathrm{HS}}, 0\}$, is also globally convergent when using an inexact line search.

4. Nonlinear least squares problems $\min_{\boldsymbol{z}} \|\boldsymbol{F}(\boldsymbol{z}, \overline{\boldsymbol{z}})\|^2$. Now we consider the special case where the objective is a nonlinear least squares problem of the form (1.2). The space spanned by $\boldsymbol{z}$ and $\overline{\boldsymbol{z}}$ is equal to $\mathcal{C}$, which is equivalent to $\mathcal{R}$ under the linear map (2.1). It is then understood that the function F can also be regarded as a function that maps $\mathcal{C}$ or $\mathcal{R}$ to $\mathbb{C}^m$, so that both the first-order model (2.7) in $\overset{\mathcal{R}}{\boldsymbol{z}}$ and the first-order model (2.8) in $\overset{\mathcal{C}}{\boldsymbol{z}}$ are applicable.

4.1. The generalized Gauss–Newton and Levenberg–Marquardt methods. The generalized Gauss–Newton and Levenberg–Marquardt methods use the first-order model

$$m_k^F(\overset{\mathcal{C}}{\boldsymbol{p}}) = F(\overset{\mathcal{C}}{\boldsymbol{z}}_k) + \frac{\partial F(\overset{\mathcal{C}}{\boldsymbol{z}}_k)}{\partial \overset{\mathcal{C}}{\boldsymbol{z}}^{\mathrm{T}}} \overset{\mathcal{C}}{\boldsymbol{p}} \tag{4.1}$$

to approximate F at the current iterate $\overset{\mathcal{C}}{\boldsymbol{z}}_k$. The objective function $f(\overset{\mathcal{C}}{\boldsymbol{z}}) \triangleq \frac{1}{2}\|F(\overset{\mathcal{C}}{\boldsymbol{z}})\|^2$ can then be approximated by

$$m_k^f(\overset{\mathcal{C}}{\boldsymbol{p}}) = \frac{1}{2}\|m_k^F(\overset{\mathcal{C}}{\boldsymbol{p}})\|^2 + \frac{\lambda_k}{2}\|\boldsymbol{p}\|^2, \tag{4.2}$$

where λ_k is the Levenberg–Marquardt regularization parameter which influences both the length and direction of the step $\boldsymbol{p}$ that minimizes m_k^f. In the Gauss–Newton method, $\lambda_k = 0$ for all k, and a trust-region framework can instead be used to control the length and direction of the step. Let $\boldsymbol{F}_k = F(\overset{\mathcal{C}}{\boldsymbol{z}}_k)$ and let $J_k = \frac{\partial F(\overset{\mathcal{C}}{\boldsymbol{z}}_k)}{\partial \overset{\mathcal{C}}{\boldsymbol{z}}^{\mathrm{T}}}$. By substituting (4.1) in (4.2), we find

$$m_k^f(\overset{\mathcal{C}}{\boldsymbol{p}}) = \frac{1}{2}\|\boldsymbol{F}_k\|^2 + \frac{1}{2}\overset{\mathcal{C}}{\boldsymbol{p}}{}^{\mathrm{T}} \begin{bmatrix} J_k S \\ \overline{J}_k \end{bmatrix}^{\mathrm{H}} \begin{bmatrix} \boldsymbol{F}_k \\ \overline{\boldsymbol{F}}_k \end{bmatrix} + \frac{1}{2}\overset{\mathcal{C}}{\boldsymbol{p}}{}^{\mathrm{H}} \left(J_k^{\mathrm{H}} J_k + \frac{\lambda_k}{2}\mathbb{I}_{2n} \right) \overset{\mathcal{C}}{\boldsymbol{p}}. \tag{4.3}$$

Comparing (4.3) to (2.10b) reveals that the complex gradient of f is given by $\frac{1}{2}(SJ_k^{\mathrm{H}}\boldsymbol{F}_k + J_k^{\mathrm{T}}\overline{\boldsymbol{F}}_k)$ and its Hessian is approximated by $J_k^{\mathrm{H}}J_k$. The conjugate complex gradient of m_k^f is given by

$$\begin{aligned} \frac{\partial m_k^f(\overset{\mathcal{C}}{\boldsymbol{p}})}{\partial \overline{\overset{\mathcal{C}}{\boldsymbol{p}}}} &= \frac{1}{2}\left(SJ_k^{\mathrm{T}}\overline{\boldsymbol{F}}_k + J_k^{\mathrm{H}}\boldsymbol{F}_k\right) + \frac{1}{2}\left(J_k^{\mathrm{H}}J_kS + SJ_k^{\mathrm{T}}\overline{J}_k + \lambda_k S\right)\overset{\mathcal{C}}{\boldsymbol{p}} \\ &= \frac{1}{2}\begin{bmatrix} J_k \\ \overline{J}_k S \end{bmatrix}^{\mathrm{H}} \begin{bmatrix} \boldsymbol{F}_k \\ \overline{\boldsymbol{F}}_k \end{bmatrix} + \frac{1}{2}\left(\begin{bmatrix} J_k \\ \overline{J}_k S \end{bmatrix}^{\mathrm{H}} \begin{bmatrix} J_k \\ \overline{J}_k S \end{bmatrix} + \lambda_k \mathbb{I}_{2n} \right) \overset{\mathcal{C}}{\boldsymbol{p}}. \end{aligned}$$

Since f is real-valued, its conjugate complex gradient must be zero when its complex gradient is zero. The minimizer of the convex quadratic model (4.2) is then

$$\overset{\mathcal{C}}{\boldsymbol{p}}{}_k^* = -\begin{bmatrix} J_k \\ \overline{J}_k S \\ \sqrt{\lambda_k}\mathbb{I}_{2n} \end{bmatrix}^\dagger \begin{bmatrix} \boldsymbol{F}_k \\ \overline{\boldsymbol{F}}_k \\ 0_{2n\times 1} \end{bmatrix}. \tag{4.4}$$

Let $A_k = \frac{\partial F(\overset{\mathcal{C}}{\boldsymbol{z}}_k)}{\partial \boldsymbol{z}^{\mathrm{T}}}$ and let $B_k = \frac{\partial F(\overset{\mathcal{C}}{\boldsymbol{z}}_k)}{\partial \overline{\boldsymbol{z}}^{\mathrm{T}}}$. A more efficient way to compute the minimizer (4.4) is

$$\overset{\mathcal{R}}{\boldsymbol{p}}{}_k^* = -\left[\begin{array}{cc} \mathrm{Re}\{A_k\}+\mathrm{Re}\{B_k\} & \mathrm{Im}\{B_k\}-\mathrm{Im}\{A_k\} \\ \mathrm{Im}\{A_k\}+\mathrm{Im}\{B_k\} & \mathrm{Re}\{A_k\}-\mathrm{Re}\{B_k\} \\ \hline \multicolumn{2}{c}{\sqrt{\lambda_k}\mathbb{I}_{2n}} \end{array}\right]^\dagger \begin{bmatrix} \mathrm{Re}\{\boldsymbol{F}_k\} \\ \mathrm{Im}\{\boldsymbol{F}_k\} \\ 0_{2n\times 1} \end{bmatrix}. \tag{4.5}$$

There are a number of cases in which additional structure in F can be exploited to simplify (4.5) further:

(i) F is analytic in $\boldsymbol{z} \in \mathbb{C}^n$; then $B_k \equiv 0_{m\times n}$:

$$\overset{\mathcal{R}}{\boldsymbol{p}}{}_k^* = -\left[\begin{array}{cc} \mathrm{Re}\{A_k\} & -\mathrm{Im}\{A_k\} \\ \mathrm{Im}\{A_k\} & \mathrm{Re}\{A_k\} \\ \hline \multicolumn{2}{c}{\sqrt{\lambda_k}\mathbb{I}_{2n}} \end{array}\right]^\dagger \begin{bmatrix} \mathrm{Re}\{\boldsymbol{F}_k\} \\ \mathrm{Im}\{\boldsymbol{F}_k\} \\ 0_{2n\times 1} \end{bmatrix} \text{ or, equivalently,}$$

$$\boldsymbol{p}_k^* = -\begin{bmatrix} A_k \\ \sqrt{\lambda_k}\mathbb{I}_n \end{bmatrix}^\dagger \begin{bmatrix} \boldsymbol{F}_k \\ 0_{n\times 1} \end{bmatrix}. \tag{4.6}$$

(ii) F is analytic in $\boldsymbol{x} \in \mathbb{R}^n$; then for $J_k \triangleq \frac{\partial F(\boldsymbol{x}_k)}{\partial \boldsymbol{x}^{\mathrm{T}}}$:

$$\boldsymbol{p}_k^* = -\left[\begin{array}{c} \mathrm{Re}\{J_k\} \\ \mathrm{Im}\{J_k\} \\ \hline \sqrt{\lambda_k}\mathbb{I}_n \end{array}\right]^\dagger \begin{bmatrix} \mathrm{Re}\{\boldsymbol{F}_k\} \\ \mathrm{Im}\{\boldsymbol{F}_k\} \\ 0_{n\times 1} \end{bmatrix}. \tag{4.7}$$

(iii) $F : \mathbb{C}^n \to \mathbb{R}^m$; then $B_k \equiv \overline{A}_k$:

$$\overset{\mathcal{R}}{\boldsymbol{p}}{}_k^* = -\left[\begin{array}{cc} 2\mathrm{Re}\{A_k\} & -2\mathrm{Im}\{A_k\} \\ \hline \multicolumn{2}{c}{\sqrt{\lambda_k}\mathbb{I}_{2n}} \end{array}\right]^\dagger \begin{bmatrix} \boldsymbol{F}_k \\ 0_{2n\times 1} \end{bmatrix}. \tag{4.8}$$

The least squares system (4.5) is equivalent to the system we would have arrived at had we applied the real Gauss–Newton or Levenberg–Marquardt methods, which is the de facto course of action for complex nonlinear least squares problems. However, expression (4.5) offers an alternative way of computing the real Jacobian using the complex Jacobian and conjugate Jacobian, which are often easier to derive and lead to more elegant expressions than is the case for the real Jacobian, as we demonstrate in the second case study. Moreover, we will also show that the real Jacobian matrix-vector product can be expressed in terms of matrix-vector products with the complex Jacobians. In this way, conjugate gradient type methods that implicitly solve the normal equations can take advantage of any structure present in the complex Jacobians.

5. Case studies.

5.1. The canonical polyadic decomposition. A tensor may be considered as an element of the tensor product of a set of vector spaces, although for this discussion their representation as a multiway array, given a choice of bases, suffices. Before we introduce the canonical polyadic decomposition, we review a few necessary definitions.

DEFINITION 5.1. *The* inner product $\langle \mathcal{T}, \mathcal{U} \rangle$ *of two tensors* $\mathcal{T}, \mathcal{U} \in \mathbb{C}^{I_1 \times \cdots \times I_N}$ *is defined as* $\langle \mathcal{T}, \mathcal{U} \rangle = \sum_{i_1=1}^{I_1} \cdots \sum_{i_N=1}^{I_N} \overline{t}_{i_1 \cdots i_N} u_{i_1 \cdots i_N}$.

DEFINITION 5.2. *The* outer product $\mathcal{T} \circ \mathcal{U}$ *of a tensor* $\mathcal{T} \in \mathbb{C}^{I_1 \times \cdots \times I_P}$ *and a tensor* $\mathcal{U} \in \mathbb{C}^{J_1 \times \cdots \times J_Q}$ *is the tensor defined by* $(\mathcal{T} \circ \mathcal{U})_{i_1 \cdots i_P j_1 \cdots j_Q} = t_{i_1 \cdots i_P} u_{j_1 \cdots j_Q}$.

DEFINITION 5.3. *An* N*th order* $\mathcal{T}$ *is* rank-one *if it is equal to the outer product of* N *nonzero vectors* $\boldsymbol{a}^{(n)} \in \mathbb{C}^{I_n}$, *i.e.,* $\mathcal{T} = \boldsymbol{a}^{(1)} \circ \cdots \circ \boldsymbol{a}^{(N)}$.

The *canonical polyadic decomposition* (CPD) [7, 18, 20, 21] approximates an Nth order tensor by a sum of R rank-one tensors (cf. Figure 5.1), i.e.,

$$\mathcal{T} \approx \sum_{r=1}^{R} \boldsymbol{a}_r^{(1)} \circ \cdots \circ \boldsymbol{a}_r^{(N)}. \tag{5.1}$$

The rank of a tensor, denoted rank($\mathcal{T}$), is defined as the minimal number of rank-one terms such that (5.1) is exact. Such a decomposition is called a rank decomposition. For any CPD, the order of the different rank-one terms may be arbitrarily permuted, and the components of the rank-one terms may be arbitrarily scaled as long as their outer product remains the same. These properties are referred to as the permutation and scaling indeterminacy, respectively. When a rank decomposition is subject only to these trivial indeterminacies, it is called essentially unique. An interesting property of higher-order tensors is that their rank decompositions are often essentially unique [25], which is not the case for matrices. Next, we introduce a few operators which will allow us to formulate the CPD as an optimization problem.

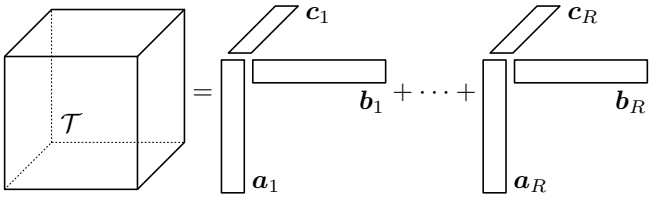

FIG. 5.1. *CPD of a third-order tensor.*

DEFINITION 5.4. *In a* mode-n matricization, flattening, *or* unfolding $T_{(n)}$ *of an* N*th order tensor* $\mathcal{T} \in \mathbb{C}^{I_1 \times \cdots \times I_N}$, *tensor element with indices* $(i_1, \ldots, i_N)$ *is mapped to matrix element* (i_n, j) *such that*

$$j = 1 + \sum_{\substack{k=1 \\ k \neq n}}^{N} (i_k - 1) J_k \text{ with } J_k = \begin{cases} 1 & \text{for } \{k = 1, k = 2 \wedge n = 1\}, \\ \prod_{\substack{m=1 \\ m \neq n}}^{k-1} I_m & \text{otherwise.} \end{cases}$$

In other words, the columns of the mode-n matricization $T_{(n)}$ are the mode-n vectors of $\mathcal{T}$ arranged along the natural order of the remaining modes.

DEFINITION 5.5. *The* Kronecker product *of two matrices* $A \in \mathbb{C}^{I \times J}$ *and* $B \in \mathbb{C}^{K \times L}$ *is defined as*

$$A \otimes B = \begin{bmatrix} a_{11}B & \cdots & a_{1J}B \\ \vdots & \ddots & \vdots \\ a_{I1}B & \cdots & a_{IJ}B \end{bmatrix}.$$

DEFINITION 5.6. *The* Khatri–Rao product [24] *of two matrices* $A \in \mathbb{C}^{I \times K}$ *and* $B \in \mathbb{C}^{J \times K}$ *is defined as* $A \odot B = \begin{bmatrix} \boldsymbol{a}_1 \otimes \boldsymbol{b}_1 & \cdots & \boldsymbol{a}_K \otimes \boldsymbol{b}_K \end{bmatrix}$.

DEFINITION 5.7. *The* Hadamard product *of two matrices* $A \in \mathbb{C}^{I\times J}$ *and* $B \in \mathbb{C}^{I\times J}$ *is the matrix defined by* $(A * B)_{ij} = a_{ij}b_{ij}$.

Let $A^{(n)} = [\boldsymbol{a}_1^{(n)} \quad \cdots \quad \boldsymbol{a}_R^{(n)}]$ be the factor matrix corresponding to the nth mode. The rank-R CPD of an Nth order tensor $\mathcal{T}$ can be written in terms of its mode-n unfoldings as

$$T_{(n)} \approx A^{(n)}(A^{(N)} \odot \cdots \odot A^{(n+1)} \odot A^{(n-1)} \odot \cdots \odot A^{(1)})^{\mathrm{T}}, \qquad n = 1, \ldots, N. \tag{5.2}$$

The CPD can now be cast as the nonlinear least squares problem

$$\min_{\substack{A^{(1)}\in\mathbb{C}^{I_1\times R}\\ \cdots \\ A^{(N)}\in\mathbb{C}^{I_N\times R}}} f_{\mathrm{CPD}} \triangleq \frac{1}{2}\|F_{\mathrm{CPD}}\|_F^2, \tag{5.3}$$

where $F_{\mathrm{CPD}} = T_{(n)} - A^{(n)}(A^{(N)} \odot \cdots \odot A^{(n+1)} \odot A^{(n-1)} \odot \cdots \odot A^{(1)})^{\mathrm{T}}$. The residuals can be arranged in N different ways, depending on which mode-n unfolding is chosen. Let $V^{\{n\}} = A^{(N)} \odot \cdots \odot A^{(n+1)} \odot A^{(n-1)} \odot \cdots \odot A^{(1)}$. The objective function f_{CPD} can be written as the sum $\frac{1}{2}(f_{\mathrm{CPD}}^{(1)} - f_{\mathrm{CPD}}^{(2)} - \overline{f}_{\mathrm{CPD}}^{(2)} + f_{\mathrm{CPD}}^{(3)})$, where

$$f_{\mathrm{CPD}}^{(1)} = \|A^{(n)}{V^{\{n\}}}^{\mathrm{T}}\|_F^2, \tag{5.4a}$$

$$f_{\mathrm{CPD}}^{(2)} = \langle T_{(n)}, A^{(n)}{V^{\{n\}}}^{\mathrm{T}}\rangle, \tag{5.4b}$$

and

$$f_{\mathrm{CPD}}^{(3)} = \|T_{(n)}\|_F^2. \tag{5.4c}$$

The partial derivative of $f_{\mathrm{CPD}}^{(1)}$ with respect to the factor matrix $A^{(n)}$ is given by

$$\begin{aligned}
\frac{\partial f_{\mathrm{CPD}}^{(1)}}{\partial A^{(n)}} &= \frac{\partial}{\partial A^{(n)}}\|A^{(n)}{V^{\{n\}}}^{\mathrm{T}}\|_F^2 \\
&= \frac{\partial}{\partial A^{(n)}}\sum_i \left(A^{(n)}\boldsymbol{v}_{i:}^{\{n\}^{\mathrm{T}}}\right)^{\mathrm{H}}\left(A^{(n)}\boldsymbol{v}_{i:}^{\{n\}^{\mathrm{T}}}\right) \\
&= \sum_i \overline{A}^{(n)}\boldsymbol{v}_{i:}^{\{n\}^{\mathrm{H}}}\boldsymbol{v}_{i:}^{\{n\}} \\
&= \overline{A}^{(n)}{V^{\{n\}}}^{\mathrm{H}}V^{\{n\}}.
\end{aligned}$$

Similarly, it is not hard to show that $\frac{\partial f_{\mathrm{CPD}}^{(2)}}{\partial A^{(n)}} = \overline{T}_{(n)}V^{\{n\}}$, $\frac{\partial \overline{f}_{\mathrm{CPD}}^{(2)}}{\partial A^{(n)}} = 0$, and $\frac{\partial f_{\mathrm{CPD}}^{(3)}}{\partial A^{(n)}} = 0$. Let

$$W^{\sigma} = \mathop{\ast}_{\substack{n=1\\ n\notin\sigma}}^{N} {A^{(n)}}^{\mathrm{H}}A^{(n)}$$

be the Hadamard product of all matrices ${A^{(n)}}^{\mathrm{H}}A^{(n)}$, $n \in \{1, \ldots, N\}\backslash\sigma$. Then a corollary of Proposition 5.8 is that ${V^{\{n\}}}^{\mathrm{H}}V^{\{n\}}$ can be evaluated more efficiently as $W^{\{n\}}$ without explicitly forming any Khatri–Rao products [4, 41].

PROPOSITION 5.8. *Let* $A^{(n)} \in \mathbb{C}^{I_n\times J}$, $n = 1, \ldots, N$, *let* $V = A^{(p_1)} \odot \cdots \odot A^{(p_N)}$ *where* $\boldsymbol{p}$ *is any permutation of* $\{1, \ldots, N\}$, *and let* $W = *_{n=1}^{N}{A^{(n)}}^{\mathrm{H}}A^{(n)}$ *be the Hadamard product of all matrices* ${A^{(n)}}^{\mathrm{H}}A^{(n)}$. *Then* $V^{\mathrm{H}}V = W$.

Proof.

$$\begin{aligned}(V^{\mathrm{H}}V)_{nm} &= (\boldsymbol{a}_n^{(p_1)} \otimes \cdots \otimes \boldsymbol{a}_n^{(p_N)})^{\mathrm{H}}(\boldsymbol{a}_m^{(p_1)} \otimes \cdots \otimes \boldsymbol{a}_m^{(p_N)}) \\ &= \langle \boldsymbol{a}_n^{(1)} \circ \cdots \circ \boldsymbol{a}_n^{(N)}, \boldsymbol{a}_m^{(1)} \circ \cdots \circ \boldsymbol{a}_m^{(N)} \rangle \\ &= (\boldsymbol{a}_n^{(1)^{\mathrm{H}}}\boldsymbol{a}_m^{(1)}) \cdots (\boldsymbol{a}_n^{(N)^{\mathrm{H}}}\boldsymbol{a}_m^{(N)}) = w_{nm}. \qquad \square\end{aligned}$$

Let $\boldsymbol{z} = \big(\mathrm{vec}(A^{(1)}), \ldots, \mathrm{vec}(A^{(N)})\big) \in \mathbb{C}^{R\sum_n^N I_n}$, where $\mathrm{vec}(\cdot)$ denotes the column-wise vectorization of a matrix. The cogradient of f_{CPD} completely determines the complex gradient of f_{CPD} and is given by

$$\frac{\partial f_{\mathrm{CPD}}}{\partial \boldsymbol{z}} = \frac{1}{2}\begin{bmatrix} \mathrm{vec}\left(\overline{A}^{(1)}W^{\{1\}} - \overline{T}_{(1)}V^{\{1\}}\right) \\ \vdots \\ \mathrm{vec}\left(\overline{A}^{(N)}W^{\{N\}} - \overline{T}_{(N)}V^{\{N\}}\right) \end{bmatrix}. \tag{5.5}$$

As outlined in section 3, it suffices to implement the function f_{CPD} and its associated scaled conjugate cogradient $\boldsymbol{g}_{\mathrm{CPD}} \triangleq 2\overline{\frac{\partial f_{\mathrm{CPD}}}{\partial \boldsymbol{z}}}$ to be able to apply any of the generalized gradient based optimization methods discussed in that section. Although we did not explicitly derive an expression for the real gradient, it is not hard to imagine that this derivation and resulting expression would be relatively more complicated compared to that of the complex gradient.

To apply either the generalized Gauss–Newton or Levenberg–Marquardt method, we define J as the Jacobian $\frac{\partial \mathrm{vec}(F_{\mathrm{CPD}})}{\partial \boldsymbol{z}^{\mathrm{T}}}$ of the nonlinear least squares problem (5.3). Since the conjugate Jacobian is identically equal to zero, F_{CPD} is analytic in $\boldsymbol{z}$ and we may use (4.6) to compute the Levenberg–Marquardt step $\boldsymbol{p}_k^*$. However, the number of rows in J is equal to $\prod_n^N I_n$ and is potentially very large compared to the number of columns $R\sum_n^N I_n$. Solving the normal equations $J_k^{\mathrm{H}}J_k\boldsymbol{p}_k = -J_k^{\mathrm{H}}\,\mathrm{vec}(F_{\mathrm{CPD}}(\boldsymbol{z}_k)) = -\boldsymbol{g}_{\mathrm{CPD}}(\boldsymbol{z}_k)$ is one way to reduce the computational complexity, although at the cost of squaring the condition number of the system. It can be shown that $J^{\mathrm{H}}J$ is an $N \times N$ block matrix, in which the (n,m)th block is the block matrix

$$\big(J^{\mathrm{H}}J\big)_{nm} = \begin{cases} W^{\{n\}} \otimes \mathbb{I}_{I_n} & \text{for } n = m, \\ \begin{bmatrix} w_{1,1}^{\{n,m\}}\boldsymbol{a}_1^{(n)}\boldsymbol{a}_1^{(m)^{\mathrm{H}}} & \cdots & w_{1,R}^{\{n,m\}}\boldsymbol{a}_R^{(n)}\boldsymbol{a}_1^{(m)^{\mathrm{H}}} \\ \vdots & \ddots & \vdots \\ w_{R,1}^{\{n,m\}}\boldsymbol{a}_1^{(n)}\boldsymbol{a}_R^{(m)^{\mathrm{H}}} & \cdots & w_{R,R}^{\{n,m\}}\boldsymbol{a}_R^{(n)}\boldsymbol{a}_R^{(m)^{\mathrm{H}}} \end{bmatrix} & \text{for } n \neq m. \end{cases} \tag{5.6}$$

Inverting $J^{\mathrm{H}}J$ costs $O((R\sum_{n=1}^N I_n)^3)$ flop unless its structure is taken into account. A straightforward way to do so is by solving the normal equations iteratively with a preconditioned conjugate gradient method. Let $Y \in \mathbb{C}^{I_m \times R}$ and let $\boldsymbol{y} = \mathrm{vec}(Y)$; then it can be shown [48] that

$$\big(J^{\mathrm{H}}J\big)_{nm}\boldsymbol{y} = \begin{cases} \mathrm{vec}\left(Y\overline{W}^{\{n\}}\right) & \text{for } n = m, \\ \mathrm{vec}\left(A^{(n)}\left(\overline{W}^{\{n,m\}} * \left(Y^{\mathrm{T}}\overline{A}^{(m)}\right)\right)\right) & \text{for } n \neq m. \end{cases} \tag{5.7}$$

The matrix-vector product $J^{\mathrm{H}}J\boldsymbol{z}$ costs only $O(R^2\sum_{n=1}^N I_n)$ flop using (5.7), which means that inverting J^HJ can be done in at most $O(R^3(\sum_{n=1}^N I_n)^2)$ flop. When $I_n \equiv I$, this is a reduction from $O(N^3R^3I^3)$ flop to $O(N^2R^3I^2)$ flop. In practice

an approximate solution can often be obtained in fewer than $R\sum_{n=1}^{N} I_n$ iterations with the help of the block-Jacobi preconditioner P, wherein the nth diagonal block is defined as $W^{\{n\}} \otimes \mathbb{I}_{I_n}$. From (5.7) we see that inverting P requires solving N square Hermitian linear systems of order R, for a total cost of $O(NR^3)$ flop. Numerical experiments comparing the above algorithms and a more detailed treatment of the CPD and its generalizations are presented in [48].

5.2. Simulation of nonlinear circuits in the frequency domain. Simulation in the frequency domain avoids many of the problems experienced when trying to use traditional time domain simulators such as SPICE [29, 40] to find the steady-state behavior of analog, radio frequency, and microwave circuits. In principle, the periodic response can always be found by integrating the differential algebraic equations that describe the system until the transient responses vanish. However, this approach can be prohibitively expensive whenever the transients are governed by large time constants. The *harmonic balance* (HB) method [30, 49] transforms these differential algebraic equations to a system of nonlinear algebraic equations in the frequency domain by representing the circuit variables by means of a truncated Fourier series of H harmonics. Because of this representation, only the steady state is simulated and its solution is furthermore limited to periodic and quasiperiodic solutions.

Consider a nonlinear circuit of $N+1$ nodes consisting of voltage-controlled current sources. Kirchoff's current law applied in the frequency domain to all but one of the nodes constitutes the set of nonlinear algebraic equations

$$I(\omega, V(\omega), \overline{V(\omega)}) = 0_{N\times 1}, \tag{5.8}$$

where $I(\omega, V(\omega), \overline{V(\omega)})$ describes the total current through each node as a function of the angular frequency ω and the node voltages $V(\omega) \in \mathbb{C}^N$. One of the nodes is selected as a reference node. The node voltages are then referenced with respect to the reference node. If the circuit were linear, (5.8) would be a parametrized linear system $Y(\omega)V(\omega) + I_s(\omega) = 0$, where $Y(\omega) \in \mathbb{C}^{N\times N}$ is a node admittance matrix and $I_s(\omega) \in \mathbb{C}^N$ represents a vector of current sources. In the HB method, the time domain voltage $v_n(t)$ of the nth node is represented by the complex coefficients $V_h^{(n)}$, $h = 0, \ldots, H-1$, in the frequency domain, i.e.,

$$v_n(t) \approx V_0^{(n)} + \sum_{h=1}^{H-1} \left(V_h^{(n)} e^{j2\pi f_0 ht} + \overline{V}_h^{(n)} e^{-j2\pi f_0 ht} \right), \tag{5.9a}$$

where

$$V_h^{(n)} = f_0 \int_0^{\frac{1}{f_0}} v_n(t) e^{-j2\pi f_0 ht} \mathrm{d}t. \tag{5.9b}$$

Analogously, the time domain current $i_n(t, v(t))$ through the nth node is a function of all node voltages $v(t) \triangleq \{v_1(t), \ldots, v_N(t)\}$ and is represented by the complex coefficients $I_h^{(n)}$, $h = 0, \ldots, H-1$, in the frequency domain.

Let $\boldsymbol{V}^{(n)} = (V_0^{(n)}, \ldots, V_{H-1}^{(n)})$ and let $\boldsymbol{I}^{(n)} = (I_0^{(n)}, \ldots, I_{H-1}^{(n)})$. The steady-state behavior of a nonlinear circuit can then be approximated as the solution of the nonlinear least squares problem

$$\min_{\substack{\boldsymbol{V}^{(1)}\in\mathbb{C}^H \\ \cdots \\ \boldsymbol{V}^{(N)}\in\mathbb{C}^H}} f_{\mathrm{HBM}} \triangleq \frac{1}{2}\|F_{\mathrm{HBM}}\|^2, \tag{5.10}$$

where $F_{\mathrm{HBM}} = (\boldsymbol{I}^{(1)}, \ldots, \boldsymbol{I}^{(N)})$. At this point the optimization problem (5.10) is usually converted to the real domain so that real optimization methods can be applied. However, the derivatives of F_{HBM} with respect to the real and imaginary parts of $V_h^{(n)}$ obscure the structure in the real Jacobian. In comparison, the expressions for the complex (conjugate) Jacobian required by the generalized Gauss–Newton and Levenberg–Marquardt methods are much more elegant. As we will see, their structure is easy to recognize and in this case also easy to exploit. The derivative of the hth harmonic of the current through the nth node $I_h^{(n)}$ with respect to the kth harmonic of the mth node voltage $V_k^{(m)}$ is

$$\begin{aligned}\frac{\partial I_h^{(n)}}{\partial V_k^{(m)}} &= f_0 \int_0^{\frac{1}{f_0}} \frac{\partial i_n(t, v(t))}{\partial v_m} \frac{\partial v_m(t)}{\partial V_k^{(m)}} e^{-j2\pi f_0 h t} \mathrm{d}t \\ &= f_0 \int_0^{\frac{1}{f_0}} \frac{\partial i_n(t, v(t))}{\partial v_m} e^{-j2\pi f_0 (h-k) t} \mathrm{d}t \triangleq G_{h-k}^{(n,m)}, \end{aligned} \tag{5.11a}$$

which is just the $(h-k)$th Fourier series coefficient of the derivative $\frac{\partial i_n(t,v(t))}{\partial v_m}$. Analogously, we have that

$$\frac{\partial I_h^{(n)}}{\partial \overline{V}_k^{(m)}} = f_0 \int_0^{\frac{1}{f_0}} \frac{\partial i_n(t, v(t))}{\partial v_m} e^{-j2\pi f_0 (h+k) t} \mathrm{d}t = G_{h+k}^{(n,m)}. \tag{5.11b}$$

Further, because $\frac{\partial i_n(t,v(t))}{\partial v_m}$ is real, $G_h^{(n,m)} = \overline{G}_{-h}^{(n,m)}$ for any integer h. Let $\boldsymbol{z} = (\boldsymbol{V}^{(1)}, \ldots, \boldsymbol{V}^{(N)})$; then the (n,m)th block of the Jacobian $\frac{\partial F_{\mathrm{HBM}}}{\partial \boldsymbol{z}^{\mathrm{T}}}$ is the Hermitian Toeplitz matrix

$$\left(\frac{\partial F_{\mathrm{HBM}}}{\partial \boldsymbol{z}^{\mathrm{T}}}\right)_{nm} = \frac{\partial \boldsymbol{I}^{(n)}}{\partial \boldsymbol{V}^{(m)\,\mathrm{T}}} = \begin{bmatrix} G_0^{(n,m)} & \overline{G}_1^{(n,m)} & \overline{G}_2^{(n,m)} & \cdots & \overline{G}_{H-1}^{(n,m)} \\ G_1^{(n,m)} & G_0^{(n,m)} & \overline{G}_1^{(n,m)} & \cdots & \overline{G}_{H-2}^{(n,m)} \\ G_2^{(n,m)} & G_1^{(n,m)} & G_0^{(n,m)} & \cdots & \overline{G}_{H-3}^{(n,m)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ G_{H-1}^{(n,m)} & G_{H-2}^{(n,m)} & G_{H-3}^{(n,m)} & \cdots & G_0^{(n,m)} \end{bmatrix}, \tag{5.12a}$$

and the (n,m)th block of the conjugate Jacobian $\frac{\partial F_{\mathrm{HBM}}}{\partial \overline{\boldsymbol{z}}^{\mathrm{T}}}$ is a Hankel matrix preceded by a zero-column, given by

$$\left(\frac{\partial F_{\mathrm{HBM}}}{\partial \overline{\boldsymbol{z}}^{\mathrm{T}}}\right)_{nm} = \frac{\partial \boldsymbol{I}^{(n)}}{\partial \overline{\boldsymbol{V}}^{(m)\,\mathrm{T}}} = \begin{bmatrix} 0 & G_1^{(n,m)} & G_2^{(n,m)} & \cdots & G_{H-1}^{(n,m)} \\ 0 & G_2^{(n,m)} & G_3^{(n,m)} & \cdots & G_H^{(n,m)} \\ 0 & G_3^{(n,m)} & G_4^{(n,m)} & \cdots & G_{H+1}^{(n,m)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & G_H^{(n,m)} & G_{H+1}^{(n,m)} & \cdots & G_{2H-2}^{(n,m)} \end{bmatrix}. \tag{5.12b}$$

For each such block, the coefficients $G_h^{(n,m)}$, $h = 0, \ldots, 2H-2$, can be approximated in $O(H \log H)$ flop using the fast Fourier transform. Assuming the average degree of the nodes in the graph associated with the circuit is constant with respect to the size of the graph, the cost of computing the (conjugate) Jacobian is $O(NH \log H)$ flop, and can be stored compactly using $O(NH)$ complex numbers. It is well known that

the product of a Toeplitz or Hankel matrix of order H with a vector can be done in $O(H \log H)$ operations [2, 23]. This property can be exploited when solving the Gauss–Newton or Levenberg–Marquardt step (4.5) iteratively. Let A_k and B_k be the Jacobian and conjugate Jacobian, respectively, and let

$$C_k = \left[\begin{array}{cc} \operatorname{Re}\{A_k\} + \operatorname{Re}\{B_k\} & \operatorname{Im}\{B_k\} - \operatorname{Im}\{A_k\} \\ \operatorname{Im}\{A_k\} + \operatorname{Im}\{B_k\} & \operatorname{Re}\{A_k\} - \operatorname{Re}\{B_k\} \\ \hline \multicolumn{2}{c}{\sqrt{\lambda_k}\mathbb{I}_{2n}} \end{array}\right]$$

be the augmented real Jacobian. Then it is not hard to show that

$$C_k \overset{\mathbb{R}}{\boldsymbol{s}} = \begin{bmatrix} \operatorname{Re}\{A_k \boldsymbol{s}\} + \operatorname{Re}\{B_k \overline{\boldsymbol{s}}\} \\ \operatorname{Im}\{A_k \boldsymbol{s}\} + \operatorname{Im}\{B_k \overline{\boldsymbol{s}}\} \\ \sqrt{\lambda_k}\operatorname{Re}\{\boldsymbol{s}\} \\ \sqrt{\lambda_k}\operatorname{Im}\{\boldsymbol{s}\} \end{bmatrix} \tag{5.13a}$$

and

$$C_k^{\mathrm{T}} \begin{bmatrix} \overset{\mathbb{R}}{\boldsymbol{t}} \\ \overset{\mathbb{R}}{\boldsymbol{u}} \end{bmatrix} = \begin{bmatrix} \operatorname{Re}\{A_k^{\mathrm{H}} \boldsymbol{t}\} + \operatorname{Re}\{B_k^{\mathrm{H}} \boldsymbol{t}\} + \sqrt{\lambda_k}\operatorname{Re}\{\boldsymbol{u}\} \\ \operatorname{Im}\{A_k^{\mathrm{H}} \boldsymbol{t}\} - \operatorname{Im}\{B_k^{\mathrm{H}} \boldsymbol{t}\} + \sqrt{\lambda_k}\operatorname{Im}\{\boldsymbol{u}\} \end{bmatrix}. \tag{5.13b}$$

The real matrix-vector products (5.13) depend only on the complex block-Toeplitz and block-Hankel matrix-vector products $A_k \boldsymbol{s}$, $B_k \overline{\boldsymbol{s}}$, $A_k^{\mathrm{H}} \boldsymbol{t}$ and $B_k^{\mathrm{H}} \boldsymbol{t}$, each of which can be computed in $O(NH \log H)$ flop. Iterative methods that implicitly solve the normal equations, such as CGLS or LSQR [19, 34], can then be applied to the real system (4.5) using the transformations (5.13). The total cost of computing the Gauss–Newton or Levenberg–Marquardt step is then at most $O(N^2 H^2 \log H)$ flop, with a memory cost of $O(NH)$ complex numbers.

If the Levenberg–Marquardt algorithm were applied in the real domain, the derivation of the first-order derivatives would not only be more involved, it would also be harder to notice and exploit their underlying structure [43]. A naive implementation would require $O(N^2 H^3)$ flop and $O(NH^2)$ memory cells to compute the step. In comparison, relatively many more harmonics can be used to represent the steady-state solution in the generalized nonlinear least squares method, so that even simulation of mildly to highly nonlinear circuits is within reach.

6. Conclusion. Functions of complex variables have several complex Taylor series expansions. The complex derivatives underlying these expansions avoid the structural overhead of dealing with derivatives of the real and imaginary parts separately. We used this approach to generalize the L-BFGS and nonlinear conjugate gradient methods to functions of complex variables. We showed that these methods need only depend on the objective function and its scaled conjugate cogradient. The latter conveniently coincides with the real gradient when optimizing over the real domain. The computational and storage cost of these generalized methods are equal to those of their real counterparts applied on the optimization problem formulated in the real domain. We also generalized Gauss–Newton and Levenberg–Marquardt methods for solving nonlinear least squares problems. We showed that the real Jacobian arising in the equivalent nonlinear least squares problem formulated in the real domain is in fact composed of a combination of two complex Jacobians. In the case studies we demonstrated that the complex gradients and complex Jacobians are not only often easier to derive than their real counterparts, but are also described by more compact

expressions. Additionally, we showed that the real Jacobian matrix-vector product can be expressed in terms ofcomplex Jacobian matrix-vector products. This allows the structure of the complex Jacobians to be exploited when applying iterative methods for solving the normal equations associated with the Gauss–Newton or Levenberg–Marquardt step. The algorithms discussed in this article, including several line search and trust-region variants, have been implemented and tested in MATLAB and are available upon request.

REFERENCES

[1] T. J. Abatzoglou, J. M. Mendel, and G. A. Harada, *The constrained total least squares technique and its applications to harmonic superresolution*, IEEE Trans. Signal Process., 39 (1991), pp. 1070–1087.

[2] Z. Bai, D. Day, J. W. Demmel, J. J. Dongarra, M. Gu, A. Ruhe, and H. A. van der Vorst, *Templates for linear algebra problems*, in Computer Science Today, J. van Leeuwen, ed., Lecture Notes in Comput. Sci. 1000, Springer, Berlin, Heidelberg, 1995, pp. 115–140.

[3] D. H. Brandwood, *A complex gradient operator and its application in adaptive array theory*, Proc. IEE-H, 130 (1983), pp. 11–16.

[4] R. Bro, *Multi-way Analysis in the Food Industry: Models, Algorithms, and Applications*, Ph.D. thesis, University of Amsterdam, Amsterdam, The Netherlands, 1998.

[5] C. G. Broyden, *The convergence of a class of double-rank minimization algorithms.* II. *The new algorithm*, J. Inst. Math. Appl., 6 (1970), pp. 222–231.

[6] R. H. Byrd, R. B. Schnabel, and G. A. Schultz, *Approximate solution of the trust region problem by minimization over two-dimensional subspaces*, Math. Programming, 40 (1988), pp. 247–263.

[7] J. D. Carroll and J.-J. Chang, *Analysis of individual differences in multidimensional scaling via an n-way generalization of "Eckart–Young" decomposition*, Psychometrika, 35 (1970), pp. 283–319.

[8] A. R. Conn, N. I. M. Gould, and Ph. L. Toint, *Trust-region methods*, MPS-SIAM Ser. Optim., SIAM and MPS, Philadelphia, 2000.

[9] J. E. Dennis, Jr., and H. H. W. Mei, *Two new unconstrained optimization algorithms which use function and gradient values*, J. Optim. Theory Appl., 28 (1979), pp. 453–482.

[10] J. E. Dennis, Jr., and J. J. Moré, *Quasi-Newton methods, motivation and theory*, SIAM Rev., 19 (1977), pp. 46–89.

[11] R. Fletcher, *A new approach to variable metric algorithms*, Comput. J., 13 (1970), pp. 317–322.

[12] R. Fletcher and C. M. Reeves, *Function minimization by conjugate gradients*, Comput. J., 7 (1964), pp. 149–154.

[13] J. C. Gilbert and C. Lemaréchal, *Some numerical experiments with variable-storage quasi-Newton algorithms*, Math. Programming, 45 (1989), pp. 407–435.

[14] J. C. Gilbert and J. Nocedal, *Global convergence properties of conjugate gradient methods for optimization*, SIAM J. Optim., 2 (1992), pp. 21–42.

[15] D. Goldfarb, *A family of variable-metric methods derived by variational means*, Math. Comp., 24 (1970), pp. 23–26.

[16] W. W. Hager and H. Zhang, *A new conjugate gradient method with guaranteed descent and an efficient line search*, SIAM J. Optim., 16 (2005), pp. 170–192.

[17] W. W. Hager and H. Zhang, *A survey of nonlinear conjugate gradient methods*, Pac. J. Optim., 2 (2006), pp. 35–58.

[18] R. A. Harshman, *Foundations of the PARAFAC procedure: Models and conditions for an "explanatory" multi-modal factor analysis*, UCLA Working Papers in Phonetics, 16 (1970), pp. 1–84.

[19] M. R. Hestenes and E. Stiefel, *Methods of conjugate gradients for solving linear systems*, J. Research Nat. Bur. Standards, 49 (1952), pp. 409–436.

[20] F. L. Hitchcock, *The expression of a tensor or a polyadic as a sum of products*, J. Math. Phys., 6 (1927), pp. 164–189.

[21] F. L. Hitchcock, *Multiple invariants and generalized rank of a p-way matrix or tensor*, J. Math. Phys., 7 (1927), pp. 39–79.

[22] R. C. James, *Advanced Calculus*, Wadsworth, Belmont, CA, 1966.

[23] T. Kailath and A. H. Sayed, eds., *Fast Reliable Algorithms for Matrices with Structure*, SIAM, Philadelphia, 1999.

[24] C. G. Khatri and C. R. Rao, *Solutions to some functional equations and their applications to characterization of probability distributions*, Sankhya Indian J. of Stat., 30 (1968), pp. 167–180.
[25] T. G. Kolda and B. W. Bader, *Tensor decompositions and applications*, SIAM Rev., 51 (2009), pp. 455–500.
[26] K. Kreutz-Delgado, *The complex gradient operator and the CR-calculus*, ArXiv e-prints, arXiv:0906.4835v1, 2009.
[27] D. C. Liu and J. Nocedal, *On the limited memory BFGS method for large scale optimization*, Math. Programming, 45 (1989), pp. 503–528.
[28] J. J. Moré and D. J. Thuente, *Line search algorithms with guaranteed sufficient decrease*, ACM Trans. Math. Software, 20 (1994), pp. 286–307.
[29] L. W. Nagel, *SPICE2: A Computer Program to Simulate Semiconductor Circuits*, Memorandum ERL-M520, Electronics Research Lab, University of California, Berkeley, CA, 1975.
[30] M. S. Nakhla and J. Vlach, *A piecewise harmonic balance technique for determination of periodic response of nonlinear systems*, IEEE Trans. Circuits and Systems, 23 (1976), pp. 85–91.
[31] Z. Nehari, *Introduction to Complex Analysis*, Allyn & Bacon, Inc., Boston, 1961.
[32] J. Nocedal, *Updating quasi-Newton matrices with limited storage*, Math. Comp., 35 (1980), pp. 773–782.
[33] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed., Springer Ser. Oper. Res., Springer-Verlag, New York, 2006.
[34] C. C. Paige and M. A. Saunders, *LSQR: An algorithm for sparse linear equations and sparse least squares*, ACM Trans. Math. Software, 8 (1982), pp. 43–71.
[35] E. Polak and G. Ribière, *Note sur la convergence de methods de directions conjugres*, Rev. Française Informat. Recherche Operationnelle, 3 (1969), pp. 35–43.
[36] M. J. D. Powell, *A hybrid method for nonlinear equations*, in Numerical Methods for Nonlinear Algebraic Equations, P. Robinowitz, ed., Gordon and Breach, London, 1970, pp. 87–144.
[37] M. J. D. Powell, *A new algorithm for unconstrained optimization*, in Nonlinear Programming, J. B. Rosen, O. L. Mangasarian, and K. Ritter, eds., Academic Press, New York, 1970, pp. 31–66.
[38] M. J. D. Powell, *Restart procedures for the conjugate gradient method*, Math. Programming, 12 (1977), pp. 241–254.
[39] M. J. D. Powell, *Nonconvex Minimization Calculations and the Conjugate Gradient Method*, Numerical Analysis, Lecture Notes in Math. 1066, Springer, Berlin, 1984.
[40] T. L. Quarles, *Analysis of Performance and Convergence Issues for Circuit Simulation*, Memorandum ERL-M89/42, Electronics Research Lab, University of California, Berkeley, CA, 1989.
[41] C. R. Rao and M. B. Rao, *Matrix Algebra and Its Applications to Statistics and Econometrics*, World Scientific, River Edge, NJ, 1998.
[42] R. Remmert, *Theory of Complex Functions*, Springer-Verlag, New York, 1991.
[43] V. Rizzoli, F. Mastri, C. Cecchetti, and F. Sgallari, *Fast and robust inexact Newton approach to the harmonic-balance analysis of nonlinear microwave circuits*, IEEE Microwave Guided Waves Lett., 7 (1997), pp. 359–361.
[44] R. B. Schnabel and J. E. Dennis, Jr., *A New Derivation of Symmetric Positive Definite Secant Updates*, Tech. report, Department of Computer Science, University of Colorado, Boulder, CO, 1980.
[45] D. F. Shanno, *Conditioning of quasi-Newton methods for function minimization*, Math. Comp., 24 (1970), pp. 647–656.
[46] J. Sherman and W. J. Morrison, *Adjustment of an inverse matrix corresponding to a change in the elements of a given column or a given row of the original matrix*, Ann. Math. Statistics, 20 (1949), p. 621.
[47] J. Sherman and W. J. Morrison, *Adjustment of an inverse matrix corresponding to a change in one element of a given matrix*, Ann. Math. Statistics, 21 (1950), pp. 124–127.
[48] L. Sorber, M. Van Barel, and L. De Lathauwer, *Optimization-based Algorithms for Tensor Decompositions: Canonical Polyadic Decomposition, Decomposition in Rank-$(L_r, L_r, 1)$ Terms and a New Generalization*, ESAT-SISTA Internal Report 12-37, Department of Electrical Engineering (ESAT), KU Leuven, Leuven, Belgium, 2012.
[49] A. Suárez, *Analysis and Design of Autonomous Microwave Circuits*, Wiley Ser. Microw. Opt. Eng. 190, Wiley-IEEE Press, Hoboken, NJ, Piscataway, NJ, 2009.
[50] A. van den Bos, *Complex gradient and Hessian*, IEE Proc. Vis. Image & Signal Process., 141 (1994), pp. 380–383.
[51] Ph. Wolfe, *Convergence conditions for ascent methods*, SIAM Rev., 11 (1969), pp. 226–235.