Abstract. While optimization is well studied for real-valued functions \( f : \mathbb{R}^N \to \mathbb{R} \), many physical problems are (partially) specified in terms of complex-valued functions \( f_c : \mathbb{C}^N \to \mathbb{C}^M \). Current optimization packages have limited support for such functions. In particular it is unclear how to define algorithmic differentiation w.r.t. complex-valued functions and arguments. This document is a collection of working notes on the topic.

Key words. First-order Methods, Algorithmic Differentiation

1. Preliminaries.

1.1. Conventions. Throughout this document, we adopt the following conventions:
- Vectors are denoted with bold lowercase letters: \( y \).
- Matrices are denoted with bold uppercase letters: \( A \).
- If \( A \in \mathbb{C}^{M \times N} \), \( a_k \in \mathbb{C}^M \) denotes the \( k \)-th column of \( A \).
- The \( i \)-th entry of vector \( y \) is denoted \( [y]_i \).
- The \((i, j)\)-th entry of matrix \( A \) is denoted \( [A]_{ij} \).
- The conjugation operator is denoted by overlining a vector or a matrix respectively: \( \bar{y}, \bar{A} \).
- The modulus of a complex number \( z \in \mathbb{C} \) is denoted by \( |z| \).
- The real/imaginary parts of matrix \( A \) are denoted \( \Re\{A\}, \Im\{A\} \), or \( A_R, A_I \), respectively.

1.2. Hadamard, Kronecker and Khatri-Rao products. The Hadamard product is the element-wise multiplication operator:

**Definition 1.1 (Hadamard product).** Let \( A \in \mathbb{C}^{M \times N} \) and \( B \in \mathbb{C}^{M \times N} \). The Hadamard product \( A \odot B \in \mathbb{C}^{M \times N} \) is defined as

\[
[A \odot B]_{ij} = [A]_{ij} [B]_{ij}.
\]

Moreover, we denote by \( A \odot^k \) the product sequence \( A \odot \cdots \odot A \).

The Kronecker product generalises the vector outer product to matrices, and represents the tensor product between two finite-dimensional linear maps:

**Definition 1.2 (Kronecker product).** Let \( A \in \mathbb{C}^{M_1 \times N_1} \) and \( B \in \mathbb{C}^{M_2 \times N_2} \). The Kronecker product \( A \otimes B \in \mathbb{C}^{M_1 M_2 \times N_1 N_2} \) is defined as

\[
(A \otimes B)_{ij} = A_{ij} B.
\]
product $A \otimes B \in \mathbb{C}^{M_1 \times N_1 \times N_2}$ is defined as

$$A \otimes B = \begin{bmatrix}
[A]_{11} B & \cdots & [A]_{1N_1} B \\
\vdots & \ddots & \vdots \\
[A]_{M_11} B & \cdots & [A]_{M_1N_1} B
\end{bmatrix}.$$ 

The main properties of the Kronecker product are [2]:

$$(1.1) \quad (A \otimes B)^H = A^H \otimes B^H,$$
$$(1.2) \quad (A \otimes B)(C \otimes D) = (AC) \otimes (BD),$$
$$(1.3) \quad (A \otimes B) \circ (C \otimes D) = (A \circ C) \otimes (B \circ D).$$

The Khatri-Rao product finally, is a column-wise Kronecker product:

**Definition 1.3 (Khatri-Rao product).** Let $A \in \mathbb{C}^{M_1 \times N}$ and $B \in \mathbb{C}^{M_2 \times N}$. The Khatri-Rao product $A \circ B \in \mathbb{C}^{M_1M_2 \times N}$ is defined as

$$A \circ B = [a_1 \otimes b_1, \ldots, a_N \otimes b_N].$$

### 1.3. Matrix identities.

$A \otimes B$ and $A \circ B$ are often too large to be stored in memory. However it is not the matrix itself that is of interest in many circumstances, but rather the effect of a linear map such as $f(x) = (A \otimes B)x$. The matrix identities below allow us to evaluate $f(x)$ without ever having to compute large intermediate arrays. They make use of the vectorisation operator:

**Definition 1.4 (Vectorisation).** Let $A \in \mathbb{C}^{M \times N}$. The vectorisation operator $\text{vec}(\cdot)$ reshapes a matrix into a vector by stacking its columns:

$$[\text{vec}(A)]_{M(j-1)+i} = [A]_{ij}.$$ 

Conversely, the matricisation operator $\text{mat}_{M,N}(\cdot)$ reshapes a vector into a matrix:

$$[\text{mat}_{M,N}(a)]_{ij} = [a]_{M(j-1)+i}.$$ 

Commonly used matrix identities are the following [1, 4]:

$$(1.4) \quad \text{vec}(ABC) = (C^T \otimes A) \text{vec}(B),$$
$$(1.5) \quad \text{vec}(A \text{ diag}(b) C) = (C^T \circ A) b,$$
$$(1.6) \quad \langle A, B \rangle_F = \text{tr} (A^H B) = \text{vec}(A)^H \text{vec}(B),$$
$$(1.7) \quad \text{vec}(ba^T) = a \otimes b.$$
The following nonstandard matrix identities are proved in Appendix A:

\[(A \circ B)^H \text{vec}(C) = \text{diag}(B^HCA)\]  
\[(A \otimes B)^H (C \otimes D) \text{vec}(E) = \text{vec}(B^HDEC^T)\]  
\[(A \circ B)^H (C \circ D) \mathbf{e} = \text{diag}(B^HD\text{diag}(e)C^T)\]  
\[(A \circ B)^H (C \circ D) = A^H C \odot B^H D\]

2. Algorithmic Differentiation. Algorithmic differentiation (AD) is an efficient procedure to evaluate numerical derivatives of mathematical expressions using a few symbolic building blocks in conjunction with the chain rule.

Definition 2.1 (Jacobian matrix). Let \(f: \mathbb{R}^N \rightarrow \mathbb{R}^M\). The Jacobian matrix \(D_f \in \mathbb{R}^{M \times N}\) is

\[
D_f = \begin{bmatrix}
\frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_N} \\
\vdots & \ddots & \vdots \\
\frac{\partial f_M}{\partial x_1} & \cdots & \frac{\partial f_M}{\partial x_N}
\end{bmatrix}.
\]

Definition 2.2 (Chain rule (real case)). Let \(f: \mathbb{R}^N \rightarrow \mathbb{R}^M\), \(g: \mathbb{R}^M \rightarrow \mathbb{R}^P\) and \(h = g \circ f\). Then

\[D_h(x) = D_g(f) D_f(x) \in \mathbb{R}^{P \times N},\]

with \(f = f(x) \in \mathbb{R}^M\).

Example 2.3. Let \(f(x) = \|y - Ax\|_2^2 = (\gamma \circ \beta \circ \alpha)(x)\), with \(\alpha: \mathbb{R}^N \rightarrow \mathbb{R}^M\), \(\beta: \mathbb{R}^M \rightarrow \mathbb{R}^M\) and \(\gamma: \mathbb{R}^M \rightarrow \mathbb{R}\).

Then \(\nabla_x f \in \mathbb{R}^{1 \times N}\) is given by

\[
\nabla_x f = D_f(x) = D_\gamma(b) D_\beta(a) D_\alpha(x)
= (2b^T)(-I_M)A
= -2b^TA,
\]

where \(a = \alpha(x) \in \mathbb{R}^M\) and \(b = \beta(a) \in \mathbb{R}^M\).

While well developed for real-valued functions \(f: \mathbb{R}^N \rightarrow \mathbb{R}^M\), generalization of Definition 2.2 to complex-valued functions \(f: \mathbb{C}^N \rightarrow \mathbb{C}^M\) is not straightforward. The generalization makes use of the hat operator:

Definition 2.4 (Hat operator). Let \(f: \mathbb{C}^N \rightarrow \mathbb{C}^M\). The hat operator \(\hat{\ }\) maps \(f\) onto its counterpart \(\hat{f}\) expressed solely in terms of real-valued expressions:
\[
f : \mathbb{C}^N \to \mathbb{C}^M
\]
\[
x_R + jx_I \to f_R(x_R + jx_I) + jf_I(x_R + jx_I)
\]
\[
\hat{f} : \mathbb{R}^{2N} \to \mathbb{R}^{2M}
\]
\[
x_R \begin{bmatrix} x_R \\ x_I \end{bmatrix} \to \begin{bmatrix} f_R(x_R, x_I) \\ f_I(x_R, x_I) \end{bmatrix}
\]

**Example 2.5 (Linear map).**
\[
f : \mathbb{C}^N \to \mathbb{C}^M
\]
\[
x_R + jx_I \to Ax
\]
\[
\hat{f} : \mathbb{R}^{2N} \to \mathbb{R}^{2M}
\]
\[
x_R \begin{bmatrix} x_R \\ x_I \end{bmatrix} \to \begin{bmatrix} A_R x_R - A_I x_I \\ A_R x_I + A_I x_R \end{bmatrix}
\]


**Definition 2.6 (Chain rule (complex case)).** Let \( f : \mathbb{C}^N \to \mathbb{C}^M, g : \mathbb{C}^M \to \mathbb{C}^P \) and \( h = g \circ f \). Then
\[
D_h(\hat{x}) = D_g(\hat{f}) \ D_f(\hat{x}) \in \mathbb{R}^{2P \times 2N}, \quad \text{with}
\]
\[
D_f(\hat{x}) = \begin{bmatrix} \frac{\partial f_R}{\partial x_R}(x_R, x_I) & \frac{\partial f_R}{\partial x_I}(x_R, x_I) \\ \frac{\partial f_I}{\partial x_R}(x_R, x_I) & \frac{\partial f_I}{\partial x_I}(x_R, x_I) \end{bmatrix},
\]

where \( f = f(x) \in \mathbb{C}^M \).

Note that the chain rule is only defined in terms of \( \hat{f} \). In particular, it is generally not possible to “unhat” \( D_f : \mathbb{R}^{2M} \to \mathbb{R}^{2N} \). However, in the special case of functions \( f : \mathbb{C}^N \to \mathbb{R}^M \), the short-hand complex-valued quantity \( D_f(x_R + jx_I) = \frac{\partial f}{\partial x_R} + j \frac{\partial f}{\partial x_I} \) is sometimes useful.

**Example 2.7.** Let \( f(x) = 1^T(y - Ax) = (\gamma \circ \beta \circ \alpha)(x) \), with
\[
\alpha : \mathbb{C}^N \to \mathbb{C}^M \\
\beta : \mathbb{C}^M \to \mathbb{C}^M \\
\gamma : \mathbb{C}^M \to \mathbb{C}
\]
\[
x \to Ax \\
a \to y - a \\
b \to 1^Tb
\]
Then \( \nabla_x \hat{f} \in \mathbb{R}^{2 \times 2N} \) is given by
\[
\nabla_x \hat{f} = D_f(\hat{x}) = D_\alpha(\hat{a}) \ D_\beta(\hat{b}) \ D_\alpha(\hat{x})
\]
\[
= \begin{bmatrix} 1_M^T & 0 \\ 0 & 1_M^T \end{bmatrix} \begin{bmatrix} -I_M & 0 \\ 0 & -I_M \end{bmatrix} \begin{bmatrix} A_R & -A_I \\ A_I & A_R \end{bmatrix}
\]
\[
= -\begin{bmatrix} 1_M A_R & -1_M A_I \\ 1_M A_I & 1_M A_R \end{bmatrix},
\]

where \( a = \alpha(x) \in \mathbb{C}^M \) and \( b = \beta(a) \in \mathbb{C}^M \). This expression cannot be further reduced to obtain a valid expression for \( \nabla_x f \).

**Example 2.8.** Let \( f(x) = \|y - Ax\|_2^2 = (\delta \circ \beta \circ \alpha)(x) \), with
\[
\alpha : \mathbb{C}^N \to \mathbb{C}^M \\
\beta : \mathbb{C}^M \to \mathbb{C}^M \\
\delta : \mathbb{C}^M \to \mathbb{R}
\]
\[
x \to Ax \\
a \to y - a \\
b \to ||b||_2^2
\]
\[ f : \mathbb{C}^N \to \mathbb{C}^M \quad \text{and} \quad D_f : \mathbb{R}^{2M} \to \mathbb{R}^{2N} \]

| \( \alpha x, \alpha \in \mathbb{C} \) | \( \begin{bmatrix} \alpha_R & -\alpha_I \\ \alpha_I & \alpha_R \end{bmatrix} \otimes I_N \) |
| \( x + y, y \in \mathbb{C}^N \) | \( I_2 \otimes I_N \) |
| \( x \) | \( \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \otimes I_N \) |
| \( Ax, A \in \mathbb{C}^{M \times N} \) | \( \begin{bmatrix} A_R & -A_I \\ A_I & A_R \end{bmatrix} \) |
| \( a \odot x, a \in \mathbb{C}^N \) | \( \begin{bmatrix} \text{diag}(a_R) - \text{diag}(a_I) \\ \text{diag}(a_I) & \text{diag}(a_R) \end{bmatrix} \) |
| \( a \odot x, a \in \mathbb{C}^K \) | \( \begin{bmatrix} a_R \otimes I_N & -a_I \otimes I_N \\ a_I \otimes I_N & a_R \otimes I_N \end{bmatrix} \) |
| \( x \odot a, a \in \mathbb{C}^K \) | \( \begin{bmatrix} I_N \otimes a_R & I_N \otimes -a_I \\ I_N \otimes a_I & I_N \otimes a_R \end{bmatrix} \) |

**Table 1**

Jacobian matrices of commonly-used operators in optimization. These can be chained using Definition 2.6 to evaluate numerical gradients of arbitrarily-complex functions.

Then \( \nabla_x \hat{f} \in \mathbb{R}^{2 \times 2N} \) is given by

\[
\nabla_x \hat{f} = D_{\hat{f}}(\hat{x}) = D_\beta(b) \ D_\beta(\hat{a}) \ D_\alpha(\hat{x}) \\
= \begin{bmatrix} 2b_R^T & 2b_I^T \\ 0 & 0 \end{bmatrix} \begin{bmatrix} -I_M & 0 \\ 0 & -I_M \end{bmatrix} \begin{bmatrix} A_R & -A_I \\ A_I & A_R \end{bmatrix} \\
= -2 \begin{bmatrix} \Re \{ b^T A \} & \Im \{ b^T A \} \\ 0 & 0 \end{bmatrix}.
\]

where \( a = \alpha(x) \in \mathbb{C}^M \) and \( b = \beta(a) \in \mathbb{C}^M \). This expression can be further reduced to obtain a valid expression for \( \nabla_x f = D_f(x) = -2b^T A \in \mathbb{C}^{1 \times N} \).

**Remark 2.9 (Implementation note).** Since optimization algorithms require (sums of) loss functions of the form \( f : \mathbb{C}^N \to \mathbb{R} \), in practice we will always be able to express gradients using the shorthand form \( \nabla_x f \in \mathbb{C}^{1 \times N} \) after applying Definition 2.6.

Table 1 provides symbolic closed-form expressions for most common operators encountered in optimization.
Appendix A. Proofs.

Proof. (1.8)

\[
[(A \odot B)^H \text{vec}(C)]_i = \langle (A \odot B)_i, \text{vec}(C) \rangle = (a_i \otimes b_i)^H \text{vec}(C)
\]

(1.7)

\[= \text{vec}(b_i a_i^T)^H \text{vec}(C)\] (1.6)

\[= \text{tr}(b_i^H C a_i) = [B^H C A]_{ii} = [\text{diag}(B^H C A)]_i \]

Proof. (1.9)

\[(A \otimes B)^H (C \otimes D) \text{vec}(E) = (A^H \otimes B^H) (C \otimes D) \text{vec}(E)\] (1.1)

(1.2)

\[= [(A^H C) \otimes (B^H D)] \text{vec}(E)\]

(1.4)

\[= \text{vec}(B^H D E C^T A)\]

Proof. (1.10)

\[(A \circ B)^H (C \circ D) e = (A \odot B)^H \text{vec}(D \text{diag}(e) C^T)\] (1.5)

(1.8)

\[= \text{diag}(B^H D \text{diag}(e) C^T A)\]

Proof. (1.11)

\[
[(A \circ B)^H (C \circ D)]_{ij} = \langle a_i \otimes b_j, c_i \otimes d_j \rangle \]

(1.7)

\[= \langle \text{vec}(b_i a_i^T), \text{vec}(d_j c_j^T) \rangle\]

(1.6)

\[= \text{tr}(b_i^H d_j c_j^T a_i) = \text{tr}(b_i^H c_j^T d_j a_i)\]

\[= \langle b_i, d_j \rangle \langle a_i, c_j \rangle .\]

When put in matrix form, the above yields

\[(A \circ B)^H (C \circ D) = A^H C \circ B^H D.\]

REFERENCES