Efficient Automatic Differentiation of Matrix Functions

Peder A. Olsen, Steven J. Rennie, and Vaibhava Goel

Abstract This paper introduces the *box product* to simplify matrix differentiation. The box product along with the Kronecker product are powerful tools to simplify matrix derivative computations for both human and machine. Moreover, they provide an abstract representation of four dimensional tensors without the cost of explicitly storing such objects. Instead the constituent matrices of the box and Kronecker products are stored, yielding a much more reasonable memory requirement. Also, the computational requirement is greatly reduced when computing the derivatives in reverse mode. Second order derivatives, and Newton-direction calculations similarly yields computational and storage benefits.

Key words: Box product, Kronecker product, Sylvester Equation, Reverse mode

1 Introduction

The computation of the derivatives of scalar functions that take a matrix argument (scalar–matrix functions) has been a topic of several publications, much of which can be found in these books, [9, 5, 11, 6]. There are also two elegant papers by Minka and Fackler that present the derivatives for many scalar–matrix functions, [7, 2]. There has even been a publication in this venue, [3]. These papers contain tables for computing the first derivative (gradient) and the second derivative (Hessian) of scalar–matrix functions $f : \mathbb{R}^{m\times n} \to \mathbb{R}$. The main tool facilitating the organized

Peder A. Olsen
IBM, TJ Watson Research Center, pederao@us.ibm.com

Steven J. Rennie
IBM, TJ Watson Research Center, sjrennie@us.ibm.com

Vaibhava Goel
IBM, TJ Watson Research Center, vgoel@us.ibm.com

computation of the second derivative is the Kronecker product. However, the task can be complex and tedious even if the function can be composed from canonical formulas listed in the previous publications. In this paper we introduce a new direct matrix product, the *box product*, that simplifies the differentiation and representation of the derivative of scalar–matrix functions. We show that the box product can be used to compactly represent the Hessian, and that the box product reveals structure that can be exploited to efficiently compute the Newton direction.

1.1 Terminology

To simplify the discussion in the following sections we will use the following terminology:

- **Scalar–Scalar Function**: A scalar function takes a scalar argument and returns a scalar: $f : \mathbb{R} \to \mathbb{R}$. An example involving matrices is $f(x) = \mathbf{a}^\top(\mathbf{Y}+x\mathbf{Z})^{-1}\mathbf{b}$.
- **Scalar–Matrix Function**: A scalar–matrix function is a function that takes a matrix argument and returns a scalar: $f : \mathbb{R}^{m\times n} \to \mathbb{R}$. An example is $f(\mathbf{X}) = \text{trace}((\mathbf{X}\Sigma\mathbf{X}^\top)^{-1})$.
- **Matrix–Matrix Function**: A matrix–matrix function is a function that takes a matrix argument and returns a scalar: $\mathbf{F} : \mathbb{R}^{m_1\times n_1} \to \mathbb{R}^{m_2\times n_2}$. An example is $\mathbf{F}(\mathbf{X}) = \mathbf{A}(\mathbf{X}\Sigma\mathbf{X}^\top)^{-1}\mathbf{B}$.

Some common unary and binary operations that yields matrix–matrix functions from matrices are: matrix multiplication, addition and subtraction, matrix inversion, transposition and scalar-multiplication of matrices. To form scalar–matrix functions we use the operations of the trace and determinant operations to go from a matrix–matrix functions to scalar–matrix functions. The trace can also be used to express the matrix inner-product, $f(\mathbf{X}) = \text{vec}^\top(\mathbf{A})\text{vec}(\mathbf{F}(\mathbf{X})) = \text{trace}(\mathbf{A}^\top\mathbf{F}(\mathbf{X}))$, and the vector-matrix-vector product, $f(\mathbf{X}) = \mathbf{a}^\top\mathbf{F}(\mathbf{X})\mathbf{b} = \text{trace}(\mathbf{F}(\mathbf{X})\mathbf{b}\mathbf{a}^\top)$. For scalar–matrix functions formed using only these operations we show how the Kronecker and Box Products helps us to compute the first and second order derivatives. For functions formed using other operations we can offer no guarantees.

The notation $\text{vec}(\mathbf{X})$ is used to indicate the column vector formed by stacking all the different columns of $\mathbf{X}$ into one big column vector.

1.2 Matrix Derivatives in Matrix Form

To compute second order derivatives of scalar–matrix functions, we first ensure that the first derivative is written in matrix form. For a scalar–matrix function we define the derivative to have the same dimensions as the matrix argument:

$$\frac{\partial f}{\partial \mathbf{X}} = \left\{ \frac{\partial f}{\partial x_{ij}} \right\}_{ij}. \tag{1}$$

For a matrix–matrix function, the derivative is a four dimensional tensor, which we reshape into a matrix as follows:

$$\frac{\partial \mathbf{F}}{\partial \mathbf{X}} \overset{\text{def}}{=} \frac{\partial \text{vec}(\mathbf{F}^\top)}{\partial \text{vec}^\top(\mathbf{X}^\top)} = \begin{pmatrix} \frac{\partial f_{11}}{\partial x_{11}} & \frac{\partial f_{11}}{\partial x_{12}} & \cdots & \frac{\partial f_{11}}{\partial x_{mn}} \\ \frac{\partial f_{12}}{\partial x_{11}} & \frac{\partial f_{12}}{\partial x_{12}} & \cdots & \frac{\partial f_{12}}{\partial x_{mn}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_{kl}}{\partial x_{11}} & \frac{\partial f_{kl}}{\partial x_{12}} & \cdots & \frac{\partial f_{kl}}{\partial x_{mn}} \end{pmatrix}. \tag{2}$$

Since a scalar–matrix function is also a 1×1 matrix–matrix function, the matrix–matrix derivative of a scalar–matrix function is according to the definition the row vector $\text{vec}^\top((\partial f/\partial \mathbf{X})^\top)$. Which form of the derivative is applied should be clear from the context. Note also that the derivative of a scalar–matrix function is a matrix–matrix function, and so the Hessian can be computed by first applying the scalar–matrix derivative followed by the matrix–matrix derivative. The Hessian of a scalar–matrix function can also be written

$$H(f(X)) = \frac{\partial^2 f}{\partial \text{vec}(\mathbf{X}^\top)\partial \text{vec}^\top(\mathbf{X}^\top)}.$$

1.3 A Simple Example

To illustrate the difficulty associated with automatically computing the derivative of a scalar–matrix function let us consider the simple example $f(\mathbf{X}) = \text{trace}(\mathbf{X}^\top \mathbf{X})$, where $\mathbf{X}^\top \in \mathbb{R}^{m \times n}$. We can compute the symbolic derivative by hand by noting that $f(\mathbf{X}) = \sum_{i=1}^m \sum_{j=1}^n x_{ij} x_{ji}$, from which it is clear that $\frac{\partial f}{\partial x_{ij}} = 2x_{ji}$ and consequently $\frac{\partial f}{\partial \mathbf{X}} = 2\mathbf{X}^\top$. Let's see what happens if we compute the function and it's gradient in forward mode. We follow [10], but use matrix terminology. First we compute the function value in forward mode:

$$\begin{aligned} \mathbf{T}_1 &= \mathbf{X}^\top && (3) \\ \mathbf{T}_2 &= \mathbf{X} && (4) \\ \mathbf{T}_3 &= \mathbf{T}_1 * \mathbf{T}_2 && (5) \\ t_4 &= \text{trace}(\mathbf{T}_3). && (6) \end{aligned}$$

Since the variables $\mathbf{T}_1, \mathbf{T}_2$ and $\mathbf{T}_3$ are matrices their derivatives are four dimensional objects that need to be stored. Let us proceed with the forward mode computation for the derivative. The matrix that permutes $\text{vec}(\mathbf{X})$ into $\text{vec}(\mathbf{X}^\top)$ is commonly known as $\mathbf{T}_{m,n}$, but we will denote it by $\mathbf{I}_m \boxtimes \mathbf{I}_n$, where $\mathbf{I}_m$ is the $m \times m$ identity matrix, and $\boxtimes$ is the box product. The expression $\mathbf{I}_m \boxtimes \mathbf{I}_n$ gives an explicit (and very useful) representation of $\mathbf{T}_{m,n}$. The general box product $\mathbf{A}\boxtimes\mathbf{B}$ is simply the result of the multiplication $(\mathbf{A}\otimes\mathbf{B})\mathbf{T}_{m,n}$. The forward mode computation of the gradient is as follows:

$$\frac{\partial \mathbf{T}_1}{\partial \mathbf{X}} = \mathbf{I}_m \boxtimes \mathbf{I}_n \tag{7}$$

$$\frac{\partial \mathbf{T}_2}{\partial \mathbf{X}} = \mathbf{I}_m \otimes \mathbf{I}_n = \mathbf{I}_{mn} \tag{8}$$

$$\frac{\partial \mathbf{T}_3}{\partial \mathbf{X}} = (\mathbf{I}_n \otimes \mathbf{T}_2^\top)\left(\frac{\partial \mathbf{T}_1}{\partial \mathbf{X}}\right) + (\mathbf{T}_1 \otimes \mathbf{I}_n)\left(\frac{\partial \mathbf{T}_2}{\partial \mathbf{X}}\right) \tag{9}$$

$$\mathrm{vec}^\top\left(\left(\frac{\partial t_4}{\partial \mathbf{X}}\right)^\top\right) = \mathrm{vec}^\top(\mathbf{I}_n)\left(\frac{\partial \mathbf{T}_3}{\partial \mathbf{X}}\right). \tag{10}$$

The total storage requirements for the matrices $\frac{\mathbf{T}_1}{\partial \mathbf{X}}$, $\frac{\mathbf{T}_2}{\partial \mathbf{X}}$, $\frac{\mathbf{T}_3}{\partial \mathbf{X}}$ and $\frac{\partial t_4}{\partial \mathbf{X}}$ is $mn + 2m^2n^2 + n^3m$, and computing $\frac{\mathbf{T}_3}{\partial \mathbf{X}}$ requires $2n^4m^2$ multiplications. To implement the derivative computation in forward mode in `C++` is a simple matter of operator overloading. Unfortunately, the resulting implementation will neither be memory nor computationally efficient.

As pointed out in [4], reverse mode computation is typically much more efficient and such is the case here as well. To avoid multiplying the large matrices we pass the $\mathrm{vec}(\mathbf{I}_n)$ in reverse to the previous steps so that all operations become matrix–vector multiplies.

$$\begin{aligned}
\mathrm{vec}^\top\left(\left(\frac{\partial t_4}{\partial \mathbf{X}}\right)^\top\right) &= \mathrm{vec}^\top(\mathbf{I}_n)\left((\mathbf{I}_n \otimes \mathbf{T}_2^\top)\left(\frac{\partial \mathbf{T}_1}{\partial \mathbf{X}}\right) + (\mathbf{T}_1 \otimes \mathbf{I}_n)\left(\frac{\partial \mathbf{T}_2}{\partial \mathbf{X}}\right)\right) && (11)\\
&= \mathrm{vec}^\top(\mathbf{I}_n\mathbf{I}_n\mathbf{T}_2)\left(\frac{\partial \mathbf{T}_1}{\partial \mathbf{X}}\right) + \mathrm{vec}^\top(\mathbf{T}_1\mathbf{I}_n\mathbf{I}_n)\left(\frac{\partial \mathbf{T}_2}{\partial \mathbf{X}}\right) && (12)\\
&= \mathrm{vec}^\top(\mathbf{T}_2)\left(\frac{\partial \mathbf{T}_1}{\partial \mathbf{X}}\right) + \mathrm{vec}^\top(\mathbf{T}_1)\left(\frac{\partial \mathbf{T}_2}{\partial \mathbf{X}}\right) && (13)\\
&= \mathrm{vec}^\top(\mathbf{T}_2)\mathbf{I}_m \boxtimes \mathbf{I}_n + \mathrm{vec}^\top(\mathbf{T}_1)\mathbf{I}_m \otimes \mathbf{I}_n && (14)\\
&= \mathrm{vec}^\top(\mathbf{T}_2^\top) + \mathrm{vec}^\top(\mathbf{T}_1) && (15)\\
&= \mathrm{vec}^\top(\mathbf{T}_1 + \mathbf{T}_2^\top). && (16)
\end{aligned}$$

To achieve the final result we did 4 matrix multiplies involving $\mathbf{I}_m$ and 4 involving $\mathbf{I}_m$. That makes for a total of $4mn(m+n)$ multiplications. Compare this to the computation of $f(\mathbf{X})$ that required mn^2 multiplications. We also see that in reverse mode we only need to store the constituent component matrices of the Kronecker and box products – thus significantly reducing the storage requirements. This example can be implemented in `C++` using expression templates and operator overloading. The overloaded matrix–matrix functions and operators record the computational graph, and the scalar–matrix function `trace` (or `det`) is overloaded to traverse the computation graph in reverse, [12].

This example was easy to work out by hand, and significantly harder using the technique described. For slightly more intricate functions the complexity can grow significantly, and the second method becomes the more practical one. Probably many of the readers will have difficulty computing the scalar–matrix derivative of $f(\mathbf{X}) = \operatorname{trace}((\mathbf{X}\Sigma\mathbf{X}^\top)^{-1})$.

In the rest of the paper we state the relevant properties of the Kronecker and box products and give all the major rules, such as the chain and product rule, necessary to compute derivatives of scalar–matrix functions functions formed from the common matrix operations.

2 Kronecker Products

For matrices $\mathbf{A} \in \mathbb{R}^{m_1 \times n_1}$ and $\mathbf{B} \in \mathbb{R}^{m_2 \times n_2}$ the Kronecker product $\mathbf{A} \otimes \mathbf{B} \in \mathbb{R}^{(m_1 m_2) \times (n_1 n_2)}$ is defined to be

$$\mathbf{A} \otimes \mathbf{B} = \begin{pmatrix} a_{11}\mathbf{B} & \cdots & a_{1n}\mathbf{B} \\ \vdots & \vdots & \vdots \\ a_{m1}\mathbf{B} & \cdots & a_{mn}\mathbf{B} \end{pmatrix}.$$

It can be verified that this definition is equivalent to $(\mathbf{A} \otimes \mathbf{B})_{(i-1)m_2+j,(k-1)n_2+l} = a_{ik}b_{jl}$, which we simply write $(\mathbf{A} \otimes \mathbf{B})_{(ij)(kl)} = a_{ik}b_{jl}$, where it is understood that the pairs ij and kl are laid out in row-major order.

Theorem 1 (Kronecker Product Identities) *Define matrices* $\mathbf{A}, \mathbf{A}_1, \mathbf{A}_2 \in \mathbb{R}^{m_1 \times n_1}$, $\mathbf{B}, \mathbf{B}_1, \mathbf{B}_2 \in \mathbb{R}^{m_2 \times n_2}$, $\mathbf{C} \in \mathbb{R}^{n_1 \times o_1}$, $\mathbf{D} \in \mathbb{R}^{n_2 \times o_2}$, $\mathbf{F} \in \mathbb{R}^{m_3 \times n_3}$ $\mathbf{X} \in \mathbb{R}^{n_1 \times m_2}$, $\mathbf{Y} \in \mathbb{R}^{n_2 \times n_1}$. $\mathbf{I}_n \in \mathbb{R}^{n \times n}$ *is used to denote the* $n \times n$ *identity matrix. The following identities hold for the Kronecker product:*

$$(\mathbf{A}_1 + \mathbf{A}_2) \otimes \mathbf{B} = \mathbf{A}_1 \otimes \mathbf{B} + \mathbf{A}_2 \otimes \mathbf{B} \tag{17}$$
$$\mathbf{A} \otimes (\mathbf{B}_1 + \mathbf{B}_2) = \mathbf{A} \otimes \mathbf{B}_1 + \mathbf{A} \otimes \mathbf{B}_2 \tag{18}$$
$$\mathbf{A} \otimes (\mathbf{B} \otimes \mathbf{F}) = (\mathbf{A} \otimes \mathbf{B}) \otimes \mathbf{F} \tag{19}$$
$$(\mathbf{A} \otimes \mathbf{B})(\mathbf{C} \otimes \mathbf{D}) = (\mathbf{AC}) \otimes (\mathbf{BD}) \tag{20}$$
$$(\mathbf{A} \otimes \mathbf{B})^\top = \mathbf{A}^\top \otimes \mathbf{B}^\top \tag{21}$$
$$(\mathbf{A} \otimes \mathbf{B})^{-1} = \mathbf{A}^{-1} \otimes \mathbf{B}^{-1} \tag{22}$$
$$\mathbf{I}_m \otimes \mathbf{I}_n = \mathbf{I}_{mn} \tag{23}$$
$$(\mathbf{A} \otimes \mathbf{B})\operatorname{vec}(\mathbf{Y}) = \operatorname{vec}(\mathbf{BYA}^\top) \tag{24}$$
$$(\mathbf{B}^\top \otimes \mathbf{A})\operatorname{vec}(\mathbf{X}) = \operatorname{vec}(\mathbf{AXB}) \tag{25}$$
$$\operatorname{trace}(\mathbf{A} \otimes \mathbf{B}) = \operatorname{trace}(\mathbf{A})\operatorname{trace}(\mathbf{B}). \tag{26}$$

If $m_1 = n_1$ *and* $m_2 = n_2$ *we also have the identity*

$$\det(\mathbf{A} \otimes \mathbf{B}) = (\det(\mathbf{A}))^{m_2} (\det(\mathbf{B}))^{m_1}. \tag{27}$$

3 Box Products

Let us first formally define the box product:

Definition 1 (Box Product) *For matrices* $\mathbf{A} \in \mathbb{R}^{m_1 \times n_1}$ *and* $\mathbf{B} \in \mathbb{R}^{m_2 \times n_2}$ *we define the box product* $\mathbf{A} \boxtimes \mathbf{B} \in \mathbb{R}^{(m_1 m_2) \times (n_1 n_2)}$ *to be*

$$(\mathbf{A} \boxtimes \mathbf{B})_{(i-1)m_2+j,(k-1)n_1+l} = a_{il} b_{jk} = (\mathbf{A} \boxtimes \mathbf{B})_{(ij)(kl)}.$$

For example, the box product of two 2×2 matrices is

$$\mathbf{A} \boxtimes \mathbf{B} = \begin{pmatrix} a_{11}b_{11} & a_{12}b_{11} & a_{11}b_{12} & a_{12}b_{12} \\ a_{11}b_{21} & a_{12}b_{21} & a_{11}b_{22} & a_{12}b_{22} \\ a_{21}b_{11} & a_{22}b_{11} & a_{21}b_{12} & a_{22}b_{12} \\ a_{21}b_{21} & a_{22}b_{21} & a_{21}b_{22} & a_{22}b_{22} \end{pmatrix}. \tag{28}$$

Theorem 2 (Box Product Identities) *Define matrices* $\mathbf{A}, \mathbf{A}_1, \mathbf{A}_2 \in \mathbb{R}^{m_1 \times n_1}$, $\mathbf{B}, \mathbf{B}_1, \mathbf{B}_2 \in \mathbb{R}^{m_2 \times n_2}$, $\mathbf{C} \in \mathbb{R}^{n_2 \times o_2}$, $\mathbf{D} \in \mathbb{R}^{n_1 \times o_1}$, $\mathbf{F} \in \mathbb{R}^{m_3 \times n_3}$ $\mathbf{X} \in \mathbb{R}^{n_1 \times m_2}$, $\mathbf{I}_n \in \mathbb{R}^{n \times n}$. *The following identities hold for the box product:*

$$(\mathbf{A}_1 + \mathbf{A}_2) \boxtimes \mathbf{B} = \mathbf{A}_1 \boxtimes \mathbf{B} + \mathbf{A}_2 \boxtimes \mathbf{B} \tag{29}$$

$$\mathbf{A} \boxtimes (\mathbf{B}_1 + \mathbf{B}_2) = \mathbf{A} \boxtimes \mathbf{B}_1 + \mathbf{A} \boxtimes \mathbf{B}_2 \tag{30}$$

Then there are the following identities that corresponds to the Kronecker case:

$$\mathbf{A} \boxtimes (\mathbf{B} \boxtimes \mathbf{F}) = (\mathbf{A} \boxtimes \mathbf{B}) \boxtimes \mathbf{F} \tag{31}$$

$$(\mathbf{A} \boxtimes \mathbf{B})(\mathbf{C} \boxtimes \mathbf{D}) = (\mathbf{A}\mathbf{D}) \otimes (\mathbf{B}\mathbf{C}) \tag{32}$$

$$(\mathbf{A} \boxtimes \mathbf{B})^\top = \mathbf{B}^\top \boxtimes \mathbf{A}^\top \tag{33}$$

$$(\mathbf{A} \boxtimes \mathbf{B})^{-1} = \mathbf{B}^{-1} \boxtimes \mathbf{A}^{-1} \tag{34}$$

$$\operatorname{trace}(\mathbf{A} \boxtimes \mathbf{B}) = \operatorname{trace}(\mathbf{A}\mathbf{B}) \tag{35}$$

In addition we have a number of identities that mix Kronecker and box-products:

$$(\mathbf{A} \boxtimes \mathbf{B})(\mathbf{C} \otimes \mathbf{D}) = (\mathbf{A}\mathbf{D}) \boxtimes (\mathbf{B}\mathbf{C}) \tag{36}$$

$$(\mathbf{A} \otimes \mathbf{B})(\mathbf{D} \boxtimes \mathbf{C}) = (\mathbf{A}\mathbf{D}) \boxtimes (\mathbf{B}\mathbf{C}) \tag{37}$$

$$(\mathbf{A} \boxtimes \mathbf{B})(\mathbf{C} \otimes \mathbf{D}) = (\mathbf{A} \otimes \mathbf{B})(\mathbf{D} \boxtimes \mathbf{C}) \tag{38}$$

$$(\mathbf{A} \boxtimes \mathbf{B})(\mathbf{C} \boxtimes \mathbf{D}) = (\mathbf{A} \otimes \mathbf{B})(\mathbf{D} \otimes \mathbf{C}) \tag{39}$$

Finally there is the `vec` *relation*

$$(\mathbf{A} \boxtimes \mathbf{B})\operatorname{vec}(\mathbf{X}) = \operatorname{vec}(\mathbf{B}\mathbf{X}^\top \mathbf{A}^\top). \tag{40}$$

3.1 Box Products of Identity Matrices

The box product of two identity matrices is a permutation matrix with many interesting properties.

Theorem 3 (Box Products of two Identity Matrices) *The Box product of two identity matrices $\mathbf{I}_m$ and $\mathbf{I}_n$ for $m,n > 1$ is a non-trivial permutation matrix $(\mathbf{I}_n \boxtimes \mathbf{I}_m \neq \mathbf{I}_{mn})$ satisfying*

$$(\mathbf{I}_m \boxtimes \mathbf{I}_n)^\top = \mathbf{I}_n \boxtimes \mathbf{I}_m \tag{41}$$
$$(\mathbf{I}_m \boxtimes \mathbf{I}_n)^\top (\mathbf{I}_m \boxtimes \mathbf{I}_n) = \mathbf{I}_{mn} \tag{42}$$
$$\det(\mathbf{I}_m \boxtimes \mathbf{I}_n) = (-1)^{mn(m-1)(n-1)/4}. \tag{43}$$

Let $\mathbf{A} \in \mathbb{R}^{m_1 \times n_1}$, $\mathbf{B} \in \mathbb{R}^{m_2 \times n_2}$. The box product of two identity matrices can be used to switch between box and Kronecker products, or to switch the order of the arguments in the box or Kronecker product.

$$(\mathbf{A} \boxtimes \mathbf{B})(\mathbf{I}_{n_2} \boxtimes \mathbf{I}_{n_1}) = (\mathbf{A} \otimes \mathbf{B}) \tag{44}$$
$$(\mathbf{A} \otimes \mathbf{B})(\mathbf{I}_{n_1} \boxtimes \mathbf{I}_{n_2}) = (\mathbf{A} \boxtimes \mathbf{B}) \tag{45}$$
$$(\mathbf{I}_{m_2} \boxtimes \mathbf{I}_{m_1})(\mathbf{A} \boxtimes \mathbf{B}) = \mathbf{B} \otimes \mathbf{A} \tag{46}$$
$$(\mathbf{I}_{m_2} \boxtimes \mathbf{I}_{m_1})(\mathbf{A} \otimes \mathbf{B}) = \mathbf{B} \boxtimes \mathbf{A} \tag{47}$$
$$(\mathbf{I}_{m_2} \boxtimes \mathbf{I}_{m_1})(\mathbf{A} \boxtimes \mathbf{B})(\mathbf{I}_{n_2} \boxtimes \mathbf{I}_{n_1}) = \mathbf{B} \boxtimes \mathbf{A} \tag{48}$$
$$(\mathbf{I}_{m_2} \boxtimes \mathbf{I}_{m_1})(\mathbf{A} \otimes \mathbf{B})(\mathbf{I}_{n_1} \boxtimes \mathbf{I}_{n_2}) = \mathbf{B} \otimes \mathbf{A} \tag{49}$$

The identity box product permutes $\mathrm{vec}(\mathbf{A})$ *into* $\mathrm{vec}(\mathbf{A}^\top)$*:*

$$(\mathbf{I}_{n_1} \boxtimes \mathbf{I}_{m_1})\mathrm{vec}(\mathbf{A}^\top) = \mathrm{vec}(\mathbf{A}) \tag{50}$$
$$(\mathbf{I}_{m_1} \boxtimes \mathbf{I}_{n_1})\mathrm{vec}(\mathbf{A}) = \mathrm{vec}(\mathbf{A}^\top) \tag{51}$$

All these box product properties are straightforward to prove, but nonetheless they are quite useful.

4 Differentiation Rules

To differentiate matrix–matrix functions there are just a few rules that are needed: Derivatives for the identity and the transpose, the product and chain rule and for the matrix inverse. We give these here without proof, and give a larger list of differentiation rules in Tables 1, 2, 3. The shorter list given here should suffice for implementation of automatic differentiation, whereas the larger list is useful for manual differentiation.

Let $\mathbf{X} \in \mathbb{R}^{m \times n}$ then the derivatives of the identity function and the transposed are

$$\frac{\partial \mathbf{X}}{\partial \mathbf{X}} = \mathbf{I}_n \otimes \mathbf{I}_m = \mathbf{I}_{mn}, \qquad \frac{\partial \mathbf{X}^\top}{\partial \mathbf{X}} = \mathbf{I}_n \boxtimes \mathbf{I}_m. \tag{52}$$

We see that transposing the variable $\mathbf{X}$ changes the Kronecker product into a box product in the expression for the derivative. This is the typical behavior. The product rule for differentiation can be concisely stated using Kronecker products. Let $\mathbf{X} \in \mathbb{R}^{m\times n}$, $\mathbf{F}(\mathbf{X}) : \mathbb{R}^{m\times n} \to \mathbb{R}^{k\times l}$, $\mathbf{G} : \mathbb{R}^{m\times n} \to \mathbb{R}^{l\times o}$ then

$$\frac{\partial \mathbf{F}(\mathbf{X})\mathbf{G}(\mathbf{X})}{\partial \mathbf{X}} = (\mathbf{I}_l \otimes \mathbf{G}^\top(\mathbf{X}))\frac{\partial \mathbf{F}(\mathbf{X})}{\partial \mathbf{X}} + (\mathbf{F}(\mathbf{X}) \otimes \mathbf{I}_l)\frac{\partial \mathbf{G}(\mathbf{X})}{\partial \mathbf{X}}. \tag{53}$$

The chain rule is simply stated as a matrix product:

$$\frac{\partial \mathbf{F}(\mathbf{G}(\mathbf{X}))}{\partial \mathbf{X}} = \left.\frac{\partial \mathbf{F}(\mathbf{Y})}{\partial \mathbf{Y}}\right|_{\mathbf{Y}=\mathbf{G}(\mathbf{X})} \frac{\partial \mathbf{G}(\mathbf{X})}{\partial \mathbf{X}}. \tag{54}$$

And finally, for square matrices, $\mathbf{X} \in \mathbb{R}^{m\times m}$ the derivatives of the matrix inverse and its transposed are

$$\frac{\partial \mathbf{X}^{-1}}{\partial \mathbf{X}} = -\mathbf{X}^{-1} \otimes \mathbf{X}^{-\top}, \qquad \frac{\partial \mathbf{X}^{-\top}}{\partial \mathbf{X}} = -\mathbf{X}^{-\top} \boxtimes \mathbf{X}^{-1}. \tag{55}$$

Scalar–matrix functions can be formed from matrix–matrix functions by use of the `trace` or `det` operations. We can take derivatives of such functions by use of the identities:

$$\mathrm{vec}\left(\left(\frac{\partial}{\partial \mathbf{X}} \log\det(\mathbf{G}(\mathbf{X}))\right)^\top\right) = \left(\frac{\partial \mathbf{G}}{\partial \mathbf{X}}\right)^\top \mathrm{vec}\left((\mathbf{G}(\mathbf{X}))^{-1}\right) \tag{56}$$

$$\mathrm{vec}\left(\left(\frac{\partial}{\partial \mathbf{X}} \mathrm{trace}(\mathbf{G}(\mathbf{X}))\right)^\top\right) = \left(\frac{\partial \mathbf{G}}{\partial \mathbf{X}}\right)^\top \mathrm{vec}(\mathbf{I}) \tag{57}$$

4.1 Hessian and Newton's Method

The methods described applied to computing the Hessian of scalar–matrix function will lead to Hessian's of the form

$$\mathbf{H} = \sum_{i=1}^{k_1} \mathrm{vec}(\mathbf{A}_i)\mathrm{vec}^\top(\mathbf{B}_i) + \sum_{i=k_1+1}^{k_2} \mathbf{A}_i \otimes \mathbf{B}_i + \sum_{i=k_2+1}^{k} \mathbf{A}_i \boxtimes \mathbf{B}_i. \tag{58}$$

For simplicity we will assume $\mathbf{A}_i, \mathbf{B}_i \in \mathbb{R}^{d\times d}$. The cost of computing $\mathbf{H}\mathrm{vec}(\mathbf{X})$ is $\mathcal{O}(kd^3)$ operations. If $k < d$ and $\mathbf{H}$ is positive definite[1] we can compute the Newton

[1] if $\mathbf{H}$ is not positive definite we can solve $\mathbf{H}^\top\mathbf{H} = \mathbf{H}^\top\nabla$ instead – doubling the computation

direction, $\mathbf{H}^{-1}\nabla$, by the conjugate gradient algorithm which uses d^2 (or less) matrix vector multiplies of the form $\mathbf{H}\mathrm{vec}(\mathbf{X})$, [8]. Thus the total computational cost is $\mathcal{O}(kd^5)$, and the memory cost is $\mathcal{O}(kd^2)$ to store $\mathbf{A}_i, \mathbf{B}_i$. For $k = 1$ or $k = 2$ the Newton direction can actually be computed in $\mathcal{O}(d^3)$ operations if the Hessian is invertible. For $k = 1$ it is clear by (22) and (34). For $k = 2$ the Newton direction is computed by use of the matrix inversion lemma if $k_1 = 1$, and otherwise ($k_1 = 0$) by transforming the Newton direction equation into a Sylvester equation. The Bartels-Stewart algorithm, [1], can then solve the Sylvester equation in $\mathcal{O}(d^3)$ operations.

4.2 An Example Taylor Series

We use these matrix differentiation rules to compute the first two terms in the Taylor series for the log-determinant. Let $f(\mathbf{X}) = \log\det(\mathbf{X})$. Then by rules (R21) and (R5) we have

$$\frac{\partial f}{\partial \mathbf{X}} = \mathbf{X}^{-\top} \qquad \frac{\partial^2 f}{\partial \mathbf{X}\partial \mathbf{X}} = \mathbf{X}^{-\top} \boxtimes \mathbf{X}^{-1}. \tag{59}$$

The Taylor series around the point $\mathbf{X}_0$ is therefore given by

$$\begin{aligned} \log\det(\mathbf{X}) &= \log\det(\mathbf{X}_0) + \mathrm{trace}((\mathbf{X}-\mathbf{X}_0)^\top \mathbf{X}_0^{-\top}) \qquad (60)\\ &\quad - \frac{1}{2!}\mathrm{vec}^\top((\mathbf{X}-\mathbf{X}_0)^\top)\left(\mathbf{X}_0^{-\top} \boxtimes \mathbf{X}_0^{-1}\right)\mathrm{vec}(\mathbf{X}-\mathbf{X}_0)^\top + \mathcal{O}((\mathbf{X}-\mathbf{X}_0)^3)\\ &= \log\det(\mathbf{X}_0) + \mathrm{trace}((\mathbf{X}-\mathbf{X}_0)\mathbf{X}_0^{-1})\\ &\quad - \frac{1}{2}\mathrm{trace}((\mathbf{X}-\mathbf{X}_0)\mathbf{X}_0^{-1}(\mathbf{X}-\mathbf{X}_0)\mathbf{X}_0^{-1}) + \mathcal{O}((\mathbf{X}-\mathbf{X}_0)^3). \end{aligned}$$

Table 1 Let $\mathbf{X} \in \mathbb{R}^{m\times m}$ be a square matrix, and $k \in \mathbb{N}$ be a positive number. The following is a list of matrix–matrix derivative rules for square matrices.

$\mathbf{F}(\mathbf{X})$	$\frac{\partial \mathbf{F}(\mathbf{X})}{\partial \mathbf{X}}$	
$\mathbf{X}$	$\mathbf{I}_{m^2}$	(R1)
$\mathbf{X}^2$	$\mathbf{I}_m \otimes \mathbf{X}^\top + \mathbf{X} \otimes \mathbf{I}_m$	(R2)
$\mathbf{X}^k$	$\sum_{i=0}^{k-1} \mathbf{X}^i \otimes \mathbf{X}^{k-1-i}$	(R3)
$\mathbf{X}^{-1}$	$-\mathbf{X}^{-1} \otimes \mathbf{X}^{-\top}$	(R4)
$\mathbf{X}^{-\top}$	$-\mathbf{X}^{-\top} \boxtimes \mathbf{X}^{-1}$	(R5)
$\mathbf{X}^{-2}$	$-\mathbf{X}^{-1} \otimes (\mathbf{X}^{-2})^\top + \mathbf{X}^{-2} \otimes \mathbf{X}^{-\top}$	(R6)
$\mathbf{X}^{-k}$	$-\sum_{i=-k}^{-1} \mathbf{X}^i \otimes (\mathbf{X}^{-k-1-i})^\top$	(R7)

Table 2 Matrix derivative rules for general matrices $\mathbf{X} \in \mathbb{R}^{m\times n}$, with $\mathbf{F}(\mathbf{X}) : \mathbb{R}^{m\times n} \to \mathbb{R}^{k\times l}$

$\mathbf{F}(\mathbf{X})$	$\frac{\partial \mathbf{F}(\mathbf{X})}{\partial \mathbf{X}}$	
$\mathbf{X}$	$\mathbf{I}_{mn}$	(R8)
$\mathbf{X}^\top$	$\mathbf{I}_n \boxtimes \mathbf{I}_m$	(R9)
$\mathbf{AX}$	$\mathbf{A} \otimes \mathbf{I}_n$	(R10)
$\mathbf{XB}$	$\mathbf{I}_m \otimes \mathbf{B}$	(R11)
$\mathbf{AXB}$	$\mathbf{A} \otimes \mathbf{B}^\top$	(R12)
$\mathbf{AX}^\top\mathbf{B}$	$\mathbf{A} \boxtimes \mathbf{B}^\top$	(R13)
$\mathbf{X}^\top\mathbf{X}$	$\mathbf{I}_n \boxtimes \mathbf{X}^\top + \mathbf{X}^\top \otimes \mathbf{I}_n$	(R14)
$\mathbf{F}(\mathbf{X}^\top)$	$\left.\frac{\partial \mathbf{F}(\mathbf{Y})}{\partial \mathbf{Y}}\right\rvert_{\mathbf{Y}=\mathbf{X}^\top} (\mathbf{I}_n \boxtimes \mathbf{I}_m)$	(R15)
$\mathbf{F}^\top(\mathbf{X})$	$(\mathbf{I}_m \boxtimes \mathbf{I}_n)\frac{\partial \mathbf{F}(\mathbf{X})}{\partial \mathbf{X}}$	(R16)
$\mathbf{AF}(\mathbf{X})\mathbf{B}$	$(\mathbf{A} \otimes \mathbf{B}^\top)\frac{\partial \mathbf{F}(\mathbf{X})}{\partial \mathbf{X}}$	(R17)
$\mathbf{F}(\mathbf{AXB})$	$\left.\frac{\partial \mathbf{F}(\mathbf{Y})}{\partial \mathbf{Y}}\right\rvert_{\mathbf{Y}=\mathbf{AXB}} (\mathbf{A} \otimes \mathbf{B}^\top)$	(R18)
$\mathbf{F}(\mathbf{G}(\mathbf{X}))$	$\left.\frac{\partial \mathbf{F}(\mathbf{Y})}{\partial \mathbf{Y}}\right\rvert_{\mathbf{Y}=\mathbf{G}(\mathbf{X})} \frac{\partial \mathbf{G}(\mathbf{X})}{\partial \mathbf{X}}$	(R19)
$\mathbf{F}(\mathbf{X})\mathbf{G}(\mathbf{X})$	$(\mathbf{I}_l \otimes \mathbf{G}^\top(\mathbf{X}))\frac{\partial \mathbf{F}(\mathbf{X})}{\partial \mathbf{X}} + (\mathbf{F}(\mathbf{X}) \otimes \mathbf{I}_l)\frac{\partial \mathbf{G}(\mathbf{X})}{\partial \mathbf{X}}$	(R20)

Table 3 Two identities useful for differentiation of scalar–matrix functions

$$\operatorname{vec}\left(\left(\frac{\partial}{\partial \mathbf{X}} \log\det(\mathbf{G}(\mathbf{X}))\right)^\top\right) = \left(\frac{\partial \mathbf{G}}{\partial \mathbf{X}}\right)^\top \operatorname{vec}\left((\mathbf{G}(\mathbf{X}))^{-1}\right) \quad \text{(R21)}$$

$$\operatorname{vec}\left(\left(\frac{\partial}{\partial \mathbf{X}} \operatorname{trace}(\mathbf{G}(\mathbf{X}))\right)^\top\right) = \left(\frac{\partial \mathbf{G}}{\partial \mathbf{X}}\right)^\top \operatorname{vec}(\mathbf{I}) \quad \text{(R22)}$$

References

1. Bartels, R., Stewart, G.: Algorithm 432: Solution of the matrix equation AX+ XB= C [F4]. Communications of the ACM **15**(9), 820–826 (1972)
2. Fackler, P.L.: Notes on matrix calculus. North Carolina State University (2005)
3. Giles, M.: Collected matrix derivative results for forward and reverse mode algorithmic differentiation. Advances in Automatic Differentiation pp. 35–44 (2008)
4. Griewank, A.: On automatic differentiation. Mathematical programming: recent developments and applications pp. 83–108 (1989)
5. Harville, D.A.: Matrix algebra from a statistician's perspective. Springer Verlag (2008)
6. Magnus, J.R., Neudecker, H.: Matrix differential calculus with applications in statistics and econometrics (revised edition). John Wiley & Sons, Ltd. (1999)
7. Minka, T.P.: Old and new matrix algebra useful for statistics. See www.stat.cmu.edu/˜minka/papers/matrix.html (2000)
8. Nocedal, J., Wright, S.: Numerical optimization. Springer Verlag (1999)
9. Petersen, K.B., Pedersen, M.S.: The matrix cookbook (2008). URL http://www2.imm.dtu.dk/pubdb/p.php?3274. Version 20081110
10. Rall, L., Corliss, G.: An introduction to automatic differentiation. Computational Differentiation: Techniques, Applications, and Tools pp. 1–17 (1996)
11. Searle, S.R.: Matrix algebra useful for statistics, vol. 512. Wiley, New York (1982)
12. Veldhuizen, T.: Expression templates. C++ Report **7**(5), 26–31 (1995)