Stokes’s theorem in R

A Preprint

Robin K. S. Hankin
Auckland University of Technology
hankin.robin@gmail.com

November 1, 2022

Abstract

In this short article I introduce the stokes package which provides functionality for working with tensors, alternating forms, wedge products, and related concepts from the exterior calculus. Notation and spirit follow Spivak. Stokes’s generalized integral theorem, viz \( \int_\partial X \phi = \int_X d\phi \), is demonstrated here using the package; it is available on CRAN at https://CRAN.R-project.org/package=stokes.

1 Introduction

Ordinary differential calculus may be formalized and generalized to arbitrary-dimensional oriented manifolds using the exterior calculus. Here I introduce the stokes package, written in the R computing language [7], which furnishes functionality for working with the exterior calculus. I give numerical verification of a number of theorems using the package. Notation follows that of Spivak [8].

2 Tensors

Recall that a \( k \)-tensor is a multilinear map \( S: V^k \to \mathbb{R} \), where \( V = \mathbb{R}^n \) is considered as a vector space; Spivak denotes the space of multilinear maps as \( \mathcal{J}^k(V) \). Formally, multilinearity means

\[
S(v_1, \ldots, av_i, \ldots, v_k) = a \cdot S(v_1, \ldots, v_i, \ldots, v_k)
\]

and

\[
S(v_1, \ldots, v_i + v_i', \ldots, v_k) = S(v_1, \ldots, v_i, \ldots, x_v) + S(v_1, \ldots, v_i', \ldots, v_k).
\]

where \( v_i \in V \) and \( a \in \mathbb{R} \). If \( S \in \mathcal{J}^k(V) \) and \( T \in \mathcal{J}^l(V) \), then we may define \( S \otimes T \in \mathcal{J}^{k+l}(V) \) as

\[
S \otimes T (v_1, \ldots, v_k, v_{k+1}, \ldots, v_{k+l}) = S(v_1, \ldots, v_k) \cdot T(v_1, \ldots, v_l)
\]

Spivak observes that \( \mathcal{J}^k(V) \) is spanned by the \( n^k \) products of the form

\[
\phi_{i_1} \otimes \phi_{i_2} \otimes \cdots \otimes \phi_{i_k} \quad 1 \leq i_1, i_2, \ldots, i_k \leq n
\]
where \( v_1, \ldots, v_k \) is a basis for \( V \) and \( \phi_i(v_j) = \delta_{ij} \); we can therefore write

\[
S = \sum_{1 \leq i_1, \ldots, i_k \leq n} a_{i_1 \ldots i_k} \phi_{i_1} \otimes \cdots \otimes \phi_{i_k}.
\] (5)

The space spanned by such products has a natural representation in \( \mathbb{R} \) as an array of dimensions \( n \times \cdots \times n = n^k \). If \( A \) is such an array, then the element \( A[i_1, i_2, \ldots, i_k] \) is the coefficient of \( \phi_{i_1} \otimes \cdots \otimes \phi_{i_k} \). However, it is more efficient and conceptually cleaner to consider a sparse array, as implemented by the \texttt{spray} package \cite{spray}.

We will consider the case \( n = 5, k = 4 \), so we have multilinear maps from \( (\mathbb{R}^5)^4 \rightarrow \mathbb{R} \). Below, we will test algebraic identities in \( \mathbb{R} \) using the idiom furnished by the \texttt{stokes} package. For our example we will define \( S = 1.5 \phi_5 \otimes \phi_1 \otimes \phi_1 \otimes \phi_1 + 2.5 \phi_1 \otimes \phi_1 \otimes \phi_2 \otimes \phi_3 + 3.5 \phi_1 \otimes \phi_3 \otimes \phi_4 \otimes \phi_2 \). Note that in some implementations the row order of object \( S \) will differ from that of \( M \); this phenomenon is due to the underlying C implementation using the \texttt{STL map} class and is discussed in more detail in the \texttt{spray} and \texttt{disordR} packages \cite{spray,disordR}.

### 2.1 Package idiom for evaluation of a tensor

First, we will define \( E \) to be a random point in \( V^k \) in terms of a matrix:

\[
> set.seed(0)
> (E <- matrix(rnorm(n*k),n,k))  # A random point in V^k
\]

Recall that \( n = 5, k = 4 \), so \( E \in (\mathbb{R}^5)^4 \). We can evaluate \( S \) at \( E \) as follows:
> f <- as.function(S)
> f(E)
[1] -3.068997

2.2 Vector space structure of tensors

Tensors have a natural vector space structure; they may be added and subtracted, and multiplied by a scalar, the same as any other vector space. Below, we define a new tensor $S_1$ and work with $2S - 3S_1$:

> $S_1$ <- as.ktensor(1+diag(4),1:4)
> 2*S-3*S1

A linear map from $V^4$ to R with $V=R^5$:

| val |
|-----|
| 5 1 1 1 = 3 |
| 1 2 1 1 = -6 |
| 1 1 2 3 =  5 |
| 1 3 4 2 =  7 |
| 1 1 1 2 = -12|
| 1 1 2 1 = - 9|
| 2 1 1 1 = - 3|

We may verify that tensors are linear using package idiom:

> LHS <- as.function(2*S-3*S1)(E)
> RHS <- 2*as.function(S)(E) -3*as.function(S1)(E)
> c(lhs=LHS,rhs=RHS,diff=LHS-RHS)

| lhs rhs diff |
|------------|---|---|
| 2.374816e+00 | 2.374816e+00 | -4.440892e-16 |

(that is, identical up to numerical precision).

2.3 Numerical verification of multilinearity in the package

Testing multilinearity is straightforward in the package. To do this, we need to define three matrices $E_1, E_2, E_3$ corresponding to points in $(R^5)^4$ which are identical except for one column. In $E_3$, this column is a linear combination of the corresponding column in $E_2$ and $E_3$:

> $E_1$ <- E
> $E_2$ <- E
> $E_3$ <- E
> x1 <- rnorm(n)
> x2 <- rnorm(n)
> r1 <- rnorm(1)
> r2 <- rnorm(1)
> E1[,2] <- x1
> E2[,2] <- x2
> E3[,2] <- r1*x1 + r2*x2

Then we can verify the multilinearity of $S$ by coercing to a function which is applied to $E_1, E_2, E_3$:

> f <- as.function(S)
> LHS <- r1*f(E1) + r2*f(E2)
> RHS <- f(E3)
> c(lhs=LHS,rhs=RHS,diff=LHS-RHS)

| lhs rhs diff |
|-------------|---|---|
| -0.5640577 | -0.5640577 | 0.0000000 |
(that is, identical up to numerical precision). Note that this is not equivalent to linearity over $V^\otimes k$:

\begin{verbatim}
> E1 <- matrix(rnorm(n*k),n,k)
> E2 <- matrix(rnorm(n*k),n,k)
> LHS <- f(r1*E1+r2*E2)
> RHS <- r1*f(E1)+r2*f(E2)
> c(lhs=LHS,rhs=RHS,diff=LHS-RHS)

lhs rhs diff
0.1731245 0.3074186 -0.1342941
\end{verbatim}

2.4 Tensor product of general tensors

Given two k-tensor objects $S,T$ we can form the tensor product $S \otimes T$, defined as

$$S \otimes T(v_1,\ldots,v_k,v_{k+1},\ldots,v_{k+l}) = S(v_1,\ldots,v_k) \cdot T(v_{k+1},\ldots,v_{k+l}) \quad (6)$$

We will calculate the tensor product of two tensors $S_1,S_2$ defined as follows:

\begin{verbatim}
> (S1 <- ktensor(spray(cbind(1:3,2:4),1:3)))
> (S2 <- as.ktensor(matrix(1:6,2,3)))
\end{verbatim}

A linear map from $V^\otimes 2$ to R with $V=R^4$:

\begin{verbatim}
val
1 2 = 1
2 3 = 2
3 4 = 3
\end{verbatim}

\begin{verbatim}
> (S2 <- as.ktensor(matrix(1:6,2,3)))
\end{verbatim}

A linear map from $V^\otimes 3$ to R with $V=R^6$:

\begin{verbatim}
val
1 3 5 = 1
2 4 6 = 1
\end{verbatim}

The R idiom for $S_1 \otimes S_2$ as per equation 6 would be `tensorprod()`, or `%x%`:

\begin{verbatim}
> tensorprod(S1,S2)
\end{verbatim}

A linear map from $V^\otimes 5$ to R with $V=R^6$:

\begin{verbatim}
val
1 2 1 3 5 = 1
3 4 1 3 5 = 3
1 2 2 4 6 = 1
3 4 2 4 6 = 3
2 3 2 4 6 = 2
2 3 1 3 5 = 2
\end{verbatim}

Then, for example:

\begin{verbatim}
> E <- matrix(rnorm(30),6,5)
> LHS <- as.function(tensorprod(S1,S2))(E)
> RHS <- as.function(S1)(E[,1:2]) * as.function(S2)(E[,3:5])
> c(lhs=LHS,rhs=RHS,diff=LHS-RHS)

lhs rhs diff
-1.048329 -1.048329 0.000000
\end{verbatim}

(that is, identical up to numerical precision).
3 Alternating forms

An alternating form is a multilinear map $T$ satisfying

$$T(v_1, \ldots, v_i, \ldots, v_j, \ldots, v_k) = -T(v_1, \ldots, v_j, \ldots, v_i, \ldots, v_k)$$

(or, equivalently, $T(v_1, \ldots, v_i, \ldots, v_i, \ldots, v_k) = 0$). We write $\Lambda^k(V)$ for the space of all alternating multilinear maps from $V^k$ to $\mathbb{R}$. Spivak gives $\text{Alt}: J^k(V) \to \Lambda^k(V)$ defined by

$$\text{Alt}(T)(v_1, \ldots, v_k) = \frac{1}{k!} \sum_{\sigma \in S_k} \text{sgn}(\sigma) \cdot T(v_{\sigma(1)}, \ldots, v_{\sigma(k)})$$

where the sum ranges over all permutations of $[n] = \{1, 2, \ldots, n\}$ and $\text{sgn}(\sigma) \in \pm 1$ is the sign of the permutation. If $T \in J^k(V)$ and $\omega \in \Lambda^k(V)$, it is straightforward to prove that $\text{Alt}(T) \in \Lambda^k(V)$, $\text{Alt}(\text{Alt}(T)) = \text{Alt}(T)$, and $\text{Alt}(\omega) = \omega$. In the stokes package, this is effected by the $\text{Alt()}$ function:

```r
> S1
A linear map from $V^2$ to $\mathbb{R}$ with $V=\mathbb{R}^4$:
  val
  1 2 = 1
  2 3 = 2
  3 4 = 3
> Alt(S1)
A linear map from $V^2$ to $\mathbb{R}$ with $V=\mathbb{R}^4$:
  val
  1 2 = 0.5
  2 1 = -0.5
  3 4 = 1.5
  2 3 = 1.0
  3 2 = -1.0
  4 3 = -1.5
```

Verifying that $\text{Alt}(S1)$ is in fact alternating is straightforward, essentially by directly evaluating Equation (8):

```r
> E <- matrix(rnorm(8),4,2)
> Erev <- E[,2:1]
> as.function(Alt(S1))(E) + as.function(Alt(S1))(Erev) # should be zero
[1] 0
```

However, we can see that this form for alternating tensors (here called $k$-forms) is inefficient and highly redundant: in this example there is a $1 \ 2$ term and a $2 \ 1$ term (the coefficients are equal and opposite). In this example we have $k = 2$ but in general there would be potentially $k!$ essentially repeated terms which collectively require only a single coefficient. The package provides $k$-form objects which are inherently alternating using a more efficient representation; they are described using wedge products which are discussed next.

3.1 Wedge products and the exterior calculus

This section follows the exposition of [5], who introduce the exterior calculus starting with a discussion of elementary forms, which are alternating forms with a particularly simple structure. An example of an
Stokes’s theorem in R

A Preprint

elementary form would be \(dx_1 \wedge dx_3\) [treated as an indivisible entity], which is an alternating multilinear map from \(\mathbb{R}^n \times \mathbb{R}^n\) to \(\mathbb{R}\) with

\[
(dx_1 \wedge dx_3) \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_n \end{pmatrix}, \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_n \end{pmatrix} = \det \begin{pmatrix} a_1 & b_1 \\ a_3 & b_3 \end{pmatrix} = a_1 b_3 - a_3 b_1
\]

That this is alternating follows from the properties of the determinant. In general of course, \(dx_i \wedge dx_j\) \(\begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix}, \begin{pmatrix} b_1 \\ b_3 \\ \vdots \\ b_n \end{pmatrix}\) = \(\det \begin{pmatrix} a_i & b_i \\ a_j & b_j \end{pmatrix}\). Because such objects are linear, it is possible to consider sums of elementary forms, such as \(dx_1 \wedge dx_2 + 3dx_2 \wedge dx_3\) with

\[
(dx_1 \wedge dx_2 + 3dx_2 \wedge dx_3) \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix}, \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix} = \det \begin{pmatrix} a_1 & b_1 \\ a_2 & b_2 \end{pmatrix} + 3\det \begin{pmatrix} a_2 & b_2 \\ a_3 & b_3 \end{pmatrix}
\]

or even \(K = dx_1 \wedge dx_2 \wedge dx_3 + 5dx_1 \wedge dx_2 \wedge dx_4\) which would be a linear map from \((\mathbb{R}^n)^3\) to \(\mathbb{R}\) with

\[
(dx_4 \wedge dx_2 \wedge dx_3 + 5dx_1 \wedge dx_2 \wedge dx_4) \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix}, \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}, \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{pmatrix} = \det \begin{pmatrix} a_4 & b_4 & c_4 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{pmatrix} + 5\det \begin{pmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_4 & b_4 & c_4 \end{pmatrix}
\]

Defining \(K\) has ready R idiom in which we define a matrix whose rows correspond to the differentials in each term:

```r
> M <- matrix(c(4,2,3,1,4,2),2,3,byrow=TRUE)
> M

[,1] [,2] [,3]
[1,] 4 2 3
[2,] 1 4 2

> K <- as.kform(M,c(1,5))
> K

An alternating linear map from \(V^3\) to \(\mathbb{R}\) with \(V=\mathbb{R}^4\):

\[
\begin{pmatrix} 2 & 3 & 4 \\ 1 & 2 & 4 \end{pmatrix} = \begin{pmatrix} 1 \\ -5 \end{pmatrix}
\]

Function \(\text{as.kform()}\) takes each row of \(M\) and places the elements in increasing order; the coefficient will change sign if the permutation is odd. Note that the order of the rows in \(K\) is immaterial and indeed in some implementations will appear in a different order: the stokes package uses the \texttt{spray} package [3] which in turn utilises the STl map class of C++; the corresponding R idiom conforms to \texttt{disordR} discipline [2].

### 3.2 Formal definition of \(dx\)

In the previous section we defined objects such as “\(dx_1 \wedge dx_6\)” as a single entity. Here I define the elementary form \(dx_i\) formally and in the next section discuss the wedge product \(\wedge\). The elementary form \(dx_i\) is simply a map from \(\mathbb{R}^n\) to \(\mathbb{R}\) with \(dx_i(x_1,x_2,\ldots,x_n) = x_i\). Observe that \(dx_i\) is an alternating form, even though we cannot swap arguments (because there is only one).
3.2.1 Package idiom for creation of differential forms

Package idiom for creating an elementary differential form appears somewhat cryptic at first sight, but is consistent (it is easier to understand package idiom for creating more complicated alternating forms, as in the next section). Suppose we wish to work with \( dx_3 \):

\[
\begin{align*}
> & \text{dx3 <- as.kform(matrix(3,1,1),1)} \\
> & \text{options(kform_symbolic_print = NULL) \# revert to default print method} \\
> & \text{dx3}
\end{align*}
\]

An alternating linear map from \( V^1 \) to \( \mathbb{R} \) with \( V=\mathbb{R}^3 \):

\[
\begin{align*}
\text{val} \\
3 & = 1
\end{align*}
\]

Interpretation of the output above is not obvious (it is easier to understand the output from more complicated alternating forms, as in the next section), but for the moment observe that \( dx_3 \) is indeed an alternating form, mapping \( \mathbb{R}^n \) to \( \mathbb{R} \) with \( dx_3(x_1, x_2, \ldots, x_n) = x_3 \). Thus, for example:

\[
\begin{align*}
> & \text{as.function(dx3)(c(14,15,16))} \\
& \quad [1] 16
\end{align*}
\]

\[
\begin{align*}
> & \text{as.function(dx3)(c(14,15,16,17,18)) \# idiom can deal with arbitrary vectors} \\
& \quad [1] 16
\end{align*}
\]

and we see that \( dx_3 \) picks out the third element of a vector. These are linear in the sense that we may add and subtract these elementary forms:

\[
\begin{align*}
> & \text{dx5 <- as.kform(matrix(5,1,1),1)} \\
> & \text{as.function(dx3 + 2*dx5)(1:10) \# picks out element 3 + 2*element 5} \\
& \quad [1] 13
\end{align*}
\]

3.3 Formal definition of wedge product

The wedge product maps two alternating forms to another alternating form; formally we write \( \wedge : \Lambda^k(V) \times \Lambda^l(V) \longrightarrow \Lambda^{k+l}(V) \). Given \( \omega \in \Lambda^k(V) \) and \( \eta \in \Lambda^l(V) \), Spivak defines the wedge product \( \omega \wedge \eta \in \Lambda^{k+l}(V) \) as

\[
\omega \wedge \eta = \left( \begin{array}{c} k+l \\ k \end{array} \right) \operatorname{Alt}(\omega \otimes \eta) \tag{12}
\]

The package includes an extensive numerically-oriented discussion of Equation 12 and its implementation in the \texttt{wedge} vignette.

3.3.1 Evaluation of the wedge product using package idiom

Wedge products are implemented in the package. To illustrate this we define two \( k \)-forms, \( K_1 \) and \( K_2 \):

\[
\begin{align*}
> & \text{(K1 <- as.kform(matrix(c(3,5,4, 4,6,1),2,3,byrow=TRUE),c(2,7)))} \\
& \text{An alternating linear map from } V^3 \text{ to } \mathbb{R} \text{ with } V=\mathbb{R}^6:} \\
& \quad \text{val} \\
& 3 4 5 = -2 \\
& 1 4 6 = 7 \\
> & \text{(K2 <- as.kform(cbind(1:5,3:7),1:5))} \\
& \text{An alternating linear map from } V^2 \text{ to } \mathbb{R} \text{ with } V=\mathbb{R}^7:} \\
& \quad \text{val}
\end{align*}
\]
\[
\begin{align*}
1 & = 1 \\
5 & = 5 \\
2 & = 2 \\
4 & = 4 \\
3 & = 3 \\
\end{align*}
\]

In symbolic notation, \(K_1\) is equal to \(7dx_1 \wedge dx_4 \wedge dx_6 - 2dx_3 \wedge dx_4 \wedge dx_5\). and \(K_2\) is \(dx_1 \wedge dx_3 + 2dx_2 \wedge dx_4 + 3dx_3 \wedge dx_5 + 4dx_4 \wedge dx_6 + 5dx_5 \wedge dx_7\). Package idiom for wedge products is straightforward; the caret (^) is overloaded to return the wedge product:

\[
> K1 ^ K2
\]

An alternating linear map from \(V^5\) to \(R\) with \(V=R^7\):

\[
\begin{align*}
val \\
1 & 4 5 6 7 = -35 \\
1 & 3 4 5 6 = -21
\end{align*}
\]

(we might write the product as \(-35dx_1 \wedge dx_4 \wedge dx_5 \wedge dx_6 \wedge dx_7 - 21dx_1 \wedge dx_3 \wedge dx_4 \wedge dx_5 \wedge dx_6\). See how the wedge product eliminates rows with repeated entries, gathers permuted rows together (respecting the sign of the permutation), and expresses the result in terms of elementary forms. The product is a linear combination of two elementary forms; note that only two coefficients out of a possible \(\binom{7}{3} = 21\) are nonzero. Note again that the order of the rows in the product is arbitrary, as per disordR discipline.

3.3.2 Associativity of the wedge product

The wedge product has formal properties such as distributivity but by far the most interesting one is associativity, which I will demonstrate below:

\[
> F1 <- as.kform(matrix(c(3,4,5, 4,6,1,3,2,1),3,3,byrow=TRUE))
> F2 <- as.kform(cbind(1:6,3:8),1:6)
> F3 <- kform_general(1:8,2)
> (F1 ^ F2) ^ F3
\]

An alternating linear map from \(V^7\) to \(R\) with \(V=R^8\):

\[
\begin{align*}
val \\
1 & 2 3 4 5 7 8 = -5 \\
1 & 3 4 5 6 7 8 = -2 \\
1 & 2 3 5 6 7 8 = 11 \\
1 & 2 3 4 5 6 8 = 1 \\
2 & 3 4 5 6 7 8 = 6 \\
1 & 2 3 4 6 7 8 = 2 \\
1 & 2 3 4 5 6 7 = 1 \\
1 & 2 4 5 6 7 8 = -5 \\
\end{align*}
\]

\[
> F1 ^ (F2 ^ F3)
\]

An alternating linear map from \(V^7\) to \(R\) with \(V=R^8\):

\[
\begin{align*}
val \\
1 & 2 3 4 5 6 7 = 1 \\
1 & 3 4 5 6 7 8 = -2 \\
1 & 2 3 4 5 6 7 8 = -5 \\
1 & 2 3 4 5 6 7 8 = 2 \\
1 & 2 3 4 5 6 8 = 1 \\
1 & 2 3 5 6 7 8 = 11 \\
2 & 3 4 5 6 7 8 = 6 \\
1 & 2 4 5 6 7 8 = -5
\end{align*}
\]

Note carefully in the above that the terms in \((F1 ^ F2) ^ F3\) and \(F1 ^ (F2 ^ F3)\) appear in a different order. They are nevertheless algebraically identical, as we may demonstrate using the (overloaded) == operator:
\[ (F_1 \wedge F_2) \wedge F_3 - F_1 \wedge (F_2 \wedge F_3) \]

The zero alternating linear map from \( V^7 \) to \( \mathbb{R} \) with \( V = \mathbb{R}^n \):
empty sparse array with 7 columns

Above we see that the two forms are identical.

### 3.4 Multilinearity of \( k \)-forms

Spivak observes that \( \Lambda^k(V) \) is spanned by the \( \binom{n}{k} \) wedge products of the form

\[ dx_{i_1} \wedge dx_{i_2} \wedge \ldots \wedge dx_{i_k} \quad 1 \leq i_1 < i_2 < \ldots < i_k \leq n \quad (13) \]

where these products are the elementary forms (compare \( J^k(V) \), which is spanned by \( n^k \) elementary forms). Formally, multilinearity means every element of the space \( \Lambda^k(V) \) is a linear combination of elementary forms, as illustrated in the package by function \texttt{kform_general}(). Consider the following idiom:

\[
\begin{align*}
> & \text{Krel <- kform_general(4,2,1:6)} \\
> & \text{Krel}
\end{align*}
\]

An alternating linear map from \( V^2 \) to \( \mathbb{R} \) with \( V = \mathbb{R}^4 \):
val
\[
\begin{align*}
1 & 2 = 1 \\
1 & 3 = 2 \\
2 & 3 = 3 \\
1 & 4 = 4 \\
2 & 4 = 5 \\
3 & 4 = 6
\end{align*}
\]

Object \texttt{Krel} is a two-form, specifically a map from \( (\mathbb{R}^4)^2 \) to \( \mathbb{R} \). Observe that \texttt{Krel} has \( \binom{4}{2} = 6 \) components, which do not appear in any particular order. Addition of such \( k \)-forms is straightforward in \( \mathbb{R} \) idiom but algebraically nontrivial:

\[
\begin{align*}
> & \text{(K1 <- as.kform(matrix(1:4,2,2),c(1,109)))} \\
> & \text{(K2 <- as.kform(matrix(c(1,3,7,8,2,4),ncol=2,byrow=TRUE),c(-1,5,4)))} \\
> & \text{(K1+K2)}
\end{align*}
\]

An alternating linear map from \( V^2 \) to \( \mathbb{R} \) with \( V = \mathbb{R}^8 \):
val
\[
\begin{align*}
1 & 3 = 1 \\
2 & 4 = 109
\end{align*}
\]

An alternating linear map from \( V^2 \) to \( \mathbb{R} \) with \( V = \mathbb{R}^8 \):
val
\[
\begin{align*}
1 & 3 = -1 \\
7 & 8 = 5 \\
2 & 4 = 4
\end{align*}
\]

Above, note how the \( dx_2 \wedge dx_4 \) terms combine [to give \( 2 \times 4 = 113 \)] and the \( dx_1 \wedge dx_3 \) term vanishes by cancellation.
3.5 Print methods

Although the spray form used above is probably the most direct and natural representation of differential forms in numerical work, sometimes we need a more algebraic print method.

```r
> U <- ktensor(spray(cbind(1:4,2:5),1:4))
> U

A linear map from V^2 to R with V=R^5:

val
1 2 = 1
2 3 = 2
3 4 = 3
4 5 = 4
```

we can represent this more algebraically using the `as.symbolic()` function:

```r
> as.symbolic(U)

[1] + a*b +2 b*c +3 c*d +4 d*e
```

In the above, `U` is a multilinear map from \((\mathbb{R}^5)^2\) to \(\mathbb{R}\). Symbolically, `a` represents the map that takes \((a,b,c,d,e)\) to `a`, `b` the map that takes \((a,b,c,d,e)\) to `b`, and so on. The asterisk `*` represents the tensor product \(\otimes\). Alternating forms work similarly but \(k\)-forms have different defaults:

```r
> K <- kform_general(3,2,1:3)
> K

An alternating linear map from V^2 to R with V=R^3:

val
1 2 = 1
1 3 = 2
2 3 = 3
```

```r
> as.symbolic(K,d="d",symbols=letters[23:26])

[1] + dw^dx +2 dw^dy +3 dx^dy
```

Note that the wedge product \(\wedge\), although implemented in package idiom as \(^\wedge\) or \(\%\%\%\), appears in the symbolic representation as an ascii caret, `^`.

We can alter the default print method with the `kform_symbolic_print` option, which uses `as.symbolic()`:

```r
> options(kform_symbolic_print = "d")
> K

An alternating linear map from V^2 to R with V=R^3:

+ dx1^dx2 +2 dx1^dx3 +3 dx2^dx3
```

This print option works nicely with the `d()` function for elementary forms:

```r
> (d(1) + d(5)) ^ (d(3)-5*d(2)) ^ d(7)

An alternating linear map from V^3 to R with V=R^7:

+ dx1^dx3^dx7 - dx3^dx5^dx7 -5 dx1^dx2^dx7 +5 dx2^dx5^dx7
```

```r
> options(kform_symbolic_print = NULL) # restore default
```

3.6 Contractions

Given a \(k\)-form \(\phi: V^k \rightarrow \mathbb{R}\) and a vector \(v \in V\), the contraction \(\phi_v\) of \(\phi\) and \(v\) is a \((k-1)\)-form with

\[
\phi_v (v^1, \ldots, v^{k-1}) = \phi(v, v^1, \ldots, v^{k-1})
\]
if \( k > 1 \); we specify \( \phi_v = \phi(v) \) if \( k = 1 \). Verification is straightforward:

> (o <- rform())  # a random 3-form

An alternating linear map from \( V^3 \) to \( R \) with \( V=R^7 \):

\[
\begin{align*}
5 & 6 & 7 = 4 \\
1 & 3 & 7 = 7 \\
2 & 3 & 7 = -2 \\
1 & 5 & 7 = -12 \\
1 & 2 & 4 = 1 \\
4 & 6 & 7 = 5 \\
2 & 4 & 6 = -6 \\
1 & 4 & 6 = 8
\end{align*}
\]

> V <- matrix(runif(21),ncol=3)
> LHS <- as.function(o)(V)
> RHS <- as.function(contract(o,V[,1]))(V[-1])
> c(LHS=LHS,RHS=RHS,diff=LHS-RHS)

\[
\begin{array}{ccc}
\text{LHS} & \text{RHS} & \text{diff} \\
4.512547e-01 & 4.512547e-01 & -4.440892e-16
\end{array}
\]

It is possible to iterate the contraction process; if we pass a matrix \( V \) to \texttt{contract()} then this is interpreted as repeated contraction with the columns of \( V \):

> as.function(contract(o,V[1:2]))(V[-(1:2),drop=FALSE])

\[ 0.4512547 \]

If we pass three columns to \texttt{contract()} the result is a 0-form:

> contract(o,V)

\[ 0.4512547 \]

In the above, the result is coerced to a scalar; in order to work with a formal 0-form (which is represented in the package as a \texttt{spray} with a zero-column index matrix) we can use the \texttt{lose=FALSE} argument:

> contract(o,V,lose=FALSE)

An alternating linear map from \( V^0 \) to \( R \) with \( V=R^0 \):

\[ 0.4512547 \]

### 3.7 Transformations and pullback

Suppose we are given a two-form \( \omega = \sum_{i<j} a_{ij} dx_i \wedge dx_j \) and relationships \( dx_i = \sum_r M_{ir} dy_r \), then we would have

\[
\omega = \sum_{i<j} a_{ij} \left( \sum_r M_{ir} dy_r \right) \wedge \left( \sum_r M_{jr} dy_r \right) .
\]

The general situation would be a \( k \)-form where we would have

\[
\omega = \sum_{i_1 < \cdots < i_k} a_{i_1 \cdots i_k} dx_{i_1} \wedge \cdots \wedge dx_{i_k}.
\]
giving

\[
\omega = \sum_{i_1 < \ldots < i_k} M_{i_1 \ldots i_k} \left( \sum_r M_{ir} dy_r \right) \wedge \cdots \wedge \left( \sum_r M_{ikr} dy_r \right).
\]

(17)

So \( \omega \) was given in terms of \( dx_1, \ldots, dx_k \) and we have expressed it in terms of \( dy_1, \ldots, dy_k \). So for example if

\[
\omega = dx_1 \wedge dx_2 + 5dx_1 \wedge dx_3
\]

(18)

and

\[
\begin{pmatrix}
dx_1 \\
dx_2 \\
dx_3
\end{pmatrix}
= \begin{pmatrix}
1 & 4 & 7 \\
2 & 5 & 8 \\
3 & 6 & 9
\end{pmatrix}
\begin{pmatrix}
dy_1 \\
dy_2 \\
dy_3
\end{pmatrix}
\]

(19)

then

\[
\omega = (1dy_1 + 4dy_2 + 7dy_3) \wedge (2dy_1 + 5dy_2 + 8dy_3) + 5(1dy_1 + 4dy_2 + 7dy_3) \wedge (3dy_1 + 6dy_2 + 9dy_3)
\]

\[
= 2dy_1 \wedge dy_2 + 5dy_1 \wedge dy_2 + \cdots + 5 \cdot 7 \cdot 6dx_3 \wedge dx_2 + 5 \cdot 7 \cdot 9dx_3 \wedge dx_3 +
\]

\[
= -33dy_1 \wedge dy_2 - 66dy_1 \wedge dy_3 - 33dy_2 \wedge dy_3
\]

(20)

Function `pullback()` does all this:

```r
> options(kform_symbolic_print = "dx")  # uses dx etc in print method
> pullback(dx^dy+5*dx^dz, matrix(1:9,3,3))
\]

An alternating linear map from \( V^2 \) to \( R \) with \( V=R^3 \):

\(-33 \ dx^\wedge dy -66 \ dx^\wedge dz -33 \ dy^\wedge dz\)

> options(kform_symbolic_print = NULL)  # revert to default

However, it is slow and I am not 100% sure that there isn’t a much more efficient way to do such a transformation. There are a few tests in tests/testthat. Here I show that transformations may be inverted using matrix inverses:

> (o <- 2 * as.kform(2) ^ as.kform(4) ^ as.kform(5))

An alternating linear map from \( V^3 \) to \( R \) with \( V=R^5 \):

val

2 4 5 = 2

> M <- matrix(rnorm(25),5,5)

Then we will transform according to matrix \( M \) and then transform according to the matrix inverse; the functionality works nicely with magrittr pipes:

> o |> pullback(M) |> pullback(solve(M))

An alternating linear map from \( V^3 \) to \( R \) with \( V=R^5 \):

val

3 4 5 = 0
1 3 4 = 0
1 2 4 = 0
1 4 5 = 0
2 3 4 = 0
2 4 5 = 2
1 3 5 = 0
2 3 5 = 0
1 2 5 = 0
Above we see many rows with values small enough for the print method to print an exact zero, but not sufficiently small to be eliminated by the `spray` internals. We can remove the small entries with `zap()`:

```r
> o |> pullback(M) |> pullback(solve(M)) |> zap()
```

An alternating linear map from $V^3$ to $R$ with $V=R^5$:

```r
tabulate(1:5)
```

See how the result is equal to the original $k$-form $2 \, dy_2 \wedge dy_4 \wedge dy_5$.

### 3.8 Exterior derivatives

Given a $k$-form $\omega$, Spivak defines the differential of $\omega$ to be a $(k+1)$-form $d\omega$ as follows. If

$$\omega = \sum_{i_1 < i_2 < \cdots < i_k} \omega_{i_1 i_2 \cdots i_k} \, dx^{i_1} \wedge dx^{i_2} \wedge \cdots \wedge dx^{i_k}$$

then

$$d\omega = \sum_{i_1 < i_2 < \cdots < i_k} \sum_{\alpha = 1}^{n} D_{i_\alpha} (\omega_{i_1 i_2 \cdots i_k}) \cdot \, dx^{i_1} \wedge dx^{i_2} \wedge \cdots \wedge dx^{i_k}$$

where $D_{i} f(a) = \lim_{h \to 0} \frac{f(a^1,\ldots,a^i+h,\ldots,a^n) - f(a^1,\ldots,a^i,\ldots,a^n)}{h}$ is the ordinary $i$th partial derivative (Spivak, p25).

This definition allows one to express the fundamental theorem of calculus in an arbitrary number of dimensions without modification. If $f: R^n \to R$ is a scalar function of position, it can be shown that

$$d(f \, dx_{i_1} \wedge \cdots \wedge dx_{i_k}) = df \wedge dx_{i_1} \wedge \cdots \wedge dx_{i_k}$$

The package provides `grad()` which, when given a vector $x_1, \ldots, x_n$ returns the one-form

$$\sum_{i=1}^{n} x_i \, dx_i$$

This is useful because $df = \sum_{j=1}^{n} (D_j f) \, dx_j$; we see that $df$ is a one-form that corresponds to the gradient $\nabla f$ of $f$ in elementary calculus: given a vector $u$ at point $p$ the correspondence would be $(\nabla f) \cdot u = df(u)$. Thus

```r
> grad(c(0.4,0.1,-3.2,1.5))
```

An alternating linear map from $V^1$ to $R$ with $V=R^4$:

```r
tabulate(c(1,2,3,4))
```

We will use the `grad()` function to verify that, in $R^n$, a certain $(k-1)$-form has zero work function. Following Hubbard and Hubbard [5], we observe that

$$F_3 = \frac{1}{(x^2 + y^2 + z^2)^{3/2}} \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

is a divergenceless velocity field in $R^3$, and thus motivated define

$$\omega_n = \frac{1}{(x_1^2 + \ldots + x_n^2)^{n/2}} \sum_{i=1}^{n} (-1)^{i-1} x_i \, dx_1 \wedge \cdots \wedge \widehat{dx_i} \wedge \cdots \wedge dx_n$$

(26)
(where a hat indicates the absence of a term). Hubbard and Hubbard show analytically that $d\omega = 0$. Here I verify this reasoning numerically, using package idiom. First we define a function that implements the wedge product $dx_1 \wedge \cdots \wedge \hat{dx}_i \wedge \cdots \wedge dx_n$:

```r
> hat <- function(x){
+   n <- length(x)
+   as.kform(t(apply(diag(n)<1,2,which)))
+ }
```

So, for example:

```r
> hat(1:5)
```

An alternating linear map from $V^4$ to $\mathbb{R}$ with $V=\mathbb{R}^5$:

```
2 3 4 5 = 1
1 3 4 5 = 1
1 2 4 5 = 1
1 2 3 5 = 1
1 2 3 4 = 1
```

Then we can use the `grad()` function to calculate $d\omega$, using the quotient law to express the derivatives analytically:

```r
> df <- function(x){
+   n <- length(x)
+   S <- sum(x^2)
+   grad(rep(c(1,-1),length=n)*(S^(n/2) - n*x^2*S^(n/2-1))/S^n
+   )
+ }
```

Thus

```r
> df(1:5)
```

An alternating linear map from $V^1$ to $\mathbb{R}$ with $V=\mathbb{R}^5$:

```
1 = 4.05e-05
2 = -2.84e-05
3 = 8.10e-06
4 = 2.03e-05
5 = -5.67e-05
```

Now we can use the wedge product of the two parts (as per equation 23) to show that the exterior derivative of $\omega_n$, evaluated at a random point in $\mathbb{R}^n$, is zero:

```r
> x <- rnorm(9)
> print(df(x) ^ hat(x)) # should be zero
```

An alternating linear map from $V^9$ to $\mathbb{R}$ with $V=\mathbb{R}^9$:

```
1 2 3 4 5 6 7 8 9 = 0
```

### 3.9 Differential of the differential, $d^2 = 0$

We can use the package to verify the celebrated fact that, for any $k$-form $\phi$, $d(d\phi) = 0$. The first step is to define scalar functions $f1()$, $f2()$, $f3()$, all 0-forms:

```r
> f1 <- function(w,x,y,z){x + y^3 + x*y*w*z}
> f2 <- function(w,x,y,z){w^2*x*y*z + sin(w) + w+z}
> f3 <- function(w,x,y,z){w*x*y*z + sin(x) + cos(w)}
```
Now we need to define elementary 1-forms:

```r
> dw <- as.kform(1)
> dx <- as.kform(2)
> dy <- as.kform(3)
> dz <- as.kform(4)
```

I will demonstrate the theorem by defining a 2-form which is the sum of three elementary two-forms, evaluated at a particular point in $\mathbb{R}^4$:

```r
> phi <-
+ (  
+   +f1(1,2,3,4) ^ dw ^ dx  
+   +f2(1,2,3,4) ^ dw ^ dy  
+   +f3(1,2,3,4) ^ dy ^ dz  
+ )
```

We could use slightly slicker R idiom by defining elementary forms $e_1,e_2,e_3$ and then defining $\phi$ to be a linear sum, weighted with 0-forms given by the (scalar) functions $f_1,f_2,f_3$:

```r
> e1 <- dw ^ dx
> e2 <- dw ^ dy
> e3 <- dy ^ dz
> phi <-
+ (  
+   +f1(1,2,3,4) ^ e1  
+   +f2(1,2,3,4) ^ e2  
+   +f3(1,2,3,4) ^ e3  
+ )
```

An alternating linear map from $V^2$ to $\mathbb{R}$ with $V=\mathbb{R}^4$:

```
val
  1 2 =  53.0000
  1 3 =  29.8414
  3 4 =  25.4496
```

Now to evaluate first derivatives of $f_1()$ etc at point $(1,2,3,4)$, using `Deriv()` from the Deriv package:

```r
> library("Deriv")
> Df1 <- Deriv(f1)(1,2,3,4)
> Df2 <- Deriv(f2)(1,2,3,4)
> Df3 <- Deriv(f3)(1,2,3,4)
```

So $Df1$ etc are numeric vectors of length 4, for example:

```r
> Df1
  w x y z
 24 13 35 6
```

To calculate $d\phi$, or $d\phi$, we can use function `grad()`:

```r
> dphi <-
+ (  
+   +grad(Df1) ^ e1  
+   +grad(Df2) ^ e2  
+   +grad(Df3) ^ e3  
+ )
```

Now we need to define elementary 1-forms:

```r
> dw <- as.kform(1)
> dx <- as.kform(2)
> dy <- as.kform(3)
> dz <- as.kform(4)
```

I will demonstrate the theorem by defining a 2-form which is the sum of three elementary two-forms, evaluated at a particular point in $\mathbb{R}^4$:

```r
> phi <-
+ (  
+   +f1(1,2,3,4) ^ dw ^ dx  
+   +f2(1,2,3,4) ^ dw ^ dy  
+   +f3(1,2,3,4) ^ dy ^ dz  
+ )
```

We could use slightly slicker R idiom by defining elementary forms $e_1,e_2,e_3$ and then defining $\phi$ to be a linear sum, weighted with 0-forms given by the (scalar) functions $f_1,f_2,f_3$:

```r
> e1 <- dw ^ dx
> e2 <- dw ^ dy
> e3 <- dy ^ dz
> phi <-
+ (  
+   +f1(1,2,3,4) ^ e1  
+   +f2(1,2,3,4) ^ e2  
+   +f3(1,2,3,4) ^ e3  
+ )
```

An alternating linear map from $V^2$ to $\mathbb{R}$ with $V=\mathbb{R}^4$:

```
val
  1 2 =  53.0000
  1 3 =  29.8414
  3 4 =  25.4496
```

Now to evaluate first derivatives of $f_1()$ etc at point $(1,2,3,4)$, using `Deriv()` from the Deriv package:

```r
> library("Deriv")
> Df1 <- Deriv(f1)(1,2,3,4)
> Df2 <- Deriv(f2)(1,2,3,4)
> Df3 <- Deriv(f3)(1,2,3,4)
```

So $Df1$ etc are numeric vectors of length 4, for example:

```r
> Df1
  w x y z
 24 13 35 6
```

To calculate $d\phi$, or $d\phi$, we can use function `grad()`:

```r
> dphi <-
+ (  
+   +grad(Df1) ^ e1  
+   +grad(Df2) ^ e2  
+   +grad(Df3) ^ e3  
+ )
```

```r
> dphi
```

```r
An alternating linear map from $V^2$ to $\mathbb{R}$ with $V=\mathbb{R}^4$:

```
val
  1 2 =  53.0000
  1 3 =  29.8414
  3 4 =  25.4496
```

Now to evaluate first derivatives of $f_1()$ etc at point $(1,2,3,4)$, using `Deriv()` from the Deriv package:

```r
> library("Deriv")
> Df1 <- Deriv(f1)(1,2,3,4)
> Df2 <- Deriv(f2)(1,2,3,4)
> Df3 <- Deriv(f3)(1,2,3,4)
```

So $Df1$ etc are numeric vectors of length 4, for example:

```r
> Df1
  w x y z
 24 13 35 6
```

To calculate $d\phi$, or $d\phi$, we can use function `grad()`:

```r
> dphi <-
+ (  
+   +grad(Df1) ^ e1  
+   +grad(Df2) ^ e2  
+   +grad(Df3) ^ e3  
+ )
```

```r
> dphi
```
An alternating linear map from $V^3$ to $\mathbb{R}$ with $V=\mathbb{R}^4$:

\[
\begin{array}{cccc}
1 & 2 & 3 & = 23.00000 \\
2 & 3 & 4 & = 11.58385 \\
1 & 2 & 4 & = 6.00000 \\
1 & 3 & 4 & = 30.15853 \\
\end{array}
\]

Now work on the differential of the differential. First evaluate the Hessians (4x4 numeric matrices) at the same point:

```r
> Hf1 <- matrix(Deriv(f1,nderiv=2)(1,2,3,4),4,4)
> Hf2 <- matrix(Deriv(f2,nderiv=2)(1,2,3,4),4,4)
> Hf3 <- matrix(Deriv(f3,nderiv=2)(1,2,3,4),4,4)
```

For example

```r
> rownames(Hf1) <- c("w","x","y","z")
> colnames(Hf1) <- c("w","x","y","z")
```

For example

```r
> Hf1
   w x y z
w  0 12  8  6
x 12  0  4  3
y  8  4 18  2
z  6  3  2  0
```

(note the matrix is symmetric; also note carefully the nonzero diagonal term). But $dd\phi$ is clearly zero as the Hessians are symmetrical:

```r
> ij <- expand.grid(seq_len(nrow(Hf1)),seq_len(ncol(Hf1)))
> ddphi <- # should be zero
+ (  
+   +as.kform(ij,c(Hf1))  
+   +as.kform(ij,c(Hf2))  
+   +as.kform(ij,c(Hf3))  
+   )
> ddphi
```

The zero alternating linear map from $V^2$ to $\mathbb{R}$ with $V=\mathbb{R}^n$:

empty sparse array with 2 columns

Above we see that $dd\phi$ is zero, as expected.

### 4 Stokes’s theorem

In its most general form, Stokes’s theorem states

\[
\int_{\partial X} \phi = \int_X d\phi
\]  
(27)

where $X \subset \mathbb{R}^n$ is a compact oriented $(k+1)$-dimensional manifold with boundary $\partial X$ and $\phi$ is a $k$-form defined on a neighborhood of $X$.

We will verify Stokes, following 6.9.5 of [5] in which

\[
\phi = \left( x_1 - x_2^2 + x_3^3 - \cdots \pm x_n^n \right) \left( \sum_{i=1}^n dx_1 \wedge \cdots \wedge \hat{dx}_i \wedge \cdots \wedge dx_n \right)  
\]  
(28)
Stokes’s theorem in R

A Preprint

(a hat indicates that a term is absent), and we wish to evaluate \( \int_{\partial C_a} \phi \) where \( C_a \) is the cube \( 0 \leq x_j \leq a, 1 \leq j \leq n \). Stokes tells us that this is equal to \( \int_{C_a} d\phi \), which is given by

\[
d\phi = (1 + 2x_2 + \cdots + nx_n^{n-1}) \, dx_1 \wedge \cdots \wedge dx_n
\]

and so the volume integral is just

\[
\sum_{j=1}^n \int_{x_1=0}^a \int_{x_2=0}^a \cdots \int_{x_i=0}^a jx_j^{j-1} \, dx_1 \, dx_2 \cdots dx_n = a^{n-1}(a + a^2 + \cdots + a^n). \tag{30}
\]

Stokes’s theorem, being trivial, is not amenable to direct numerical verification but the package does allow slick creation of \( \phi \):

\[
> \text{phi} <- \text{function}(x)\{
+ n <- \text{length}(x)
+ \text{sum}(x^{\text{seq_len}(n)}*\text{rep_len(c(1,-1),n)}) * \text{as.kform(t(apply(diag(n)<1,2,which)))}
+ \}
> \text{phi}(1:9)
\]

An alternating linear map from \( V^8 \) to \( R \) with \( V=R^9 \):

\[
\begin{align*}
1 & 2 & 3 & 4 & 6 & 7 & 8 & 9 & = 371423053 \\
1 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & = 371423053 \\
1 & 2 & 3 & 4 & 5 & 6 & 7 & 9 & = 371423053 \\
1 & 2 & 3 & 4 & 5 & 7 & 8 & 9 & = 371423053 \\
1 & 2 & 4 & 5 & 6 & 7 & 8 & 9 & = 371423053 \\
2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & = 371423053 \\
1 & 2 & 3 & 4 & 5 & 6 & 8 & 9 & = 371423053 \\
1 & 2 & 3 & 5 & 6 & 7 & 8 & 9 & = 371423053 \\
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & = 371423053
\end{align*}
\]

Recall that \( \phi \) is a function that maps \( \mathbb{R}^9 \) to \( 8 \)-forms. Here we choose \( (1, 2, \ldots, 9) \in \mathbb{R}^9 \) and \( \text{phi}(1:9) \) as shown above is the resulting \( 8 \)-form. Thus, if we write \( \phi_{1:9} \) for \( \text{phi}(1:9) \) we would have \( \phi_{1:9} : (\mathbb{R}^9)^8 \longrightarrow \mathbb{R} \), with package idiom as follows:

\[
> E <- \text{matrix(\text{runif(72),9,8})} \quad \# (R^9)^8
> \text{as.function(\text{phi}(1:9))(E)}
\]

[1] -26620528

Further, \( d\phi \) is given by

\[
> \text{dphi} <- \text{function}(x)\{
+ \text{nn} <- \text{seq_along}(x)
+ \text{sum}(\text{nn}*\text{seq}(\text{nn}-1)) * \text{as.kform(seq_along(x))}
+ \}
> \text{dphi}(1:9)
\]

An alternating linear map from \( V^9 \) to \( R \) with \( V=R^9 \):

\[
\begin{align*}
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & = 405071317 \\
1 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & = 405071317 \\
1 & 2 & 3 & 4 & 5 & 6 & 7 & 9 & = 405071317 \\
1 & 2 & 3 & 4 & 5 & 7 & 8 & 9 & = 405071317 \\
1 & 2 & 4 & 5 & 6 & 7 & 8 & 9 & = 405071317 \\
2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & = 405071317 \\
1 & 2 & 3 & 4 & 5 & 6 & 8 & 9 & = 405071317 \\
1 & 2 & 3 & 5 & 6 & 7 & 8 & 9 & = 405071317 \\
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & = 405071317
\end{align*}
\]

Observe that \( \text{dphi}(1:9) \) is a \( 9 \)-form, with \( d\phi_{1:9} : (\mathbb{R}^9)^9 \longrightarrow \mathbb{R} \). Now consider Spivak’s theorem 4.6 (page 82), which in this context states that a \( 9 \)-form is proportional to the determinant of the \( 9 \times 9 \) matrix formed from its arguments, with constant of proportionality equal to the form evaluated on the identity matrix \( I_9 \) [formally and more generally, if \( v_1, \ldots, v_n \) is a basis for \( V \), \( \omega \in \Lambda^n(V) \) and \( w_i = \sum a_{ij} v_j \) then \( \omega(w_1, \ldots, w_n) = \det (a_{ij}) \cdot \omega(v_1, \ldots, v_n) \)]. Numerically:
> f <- as.function(dphi(1:9))
> E <- matrix(runif(81),9,9)
> LHS <- f(E)
> RHS <- det(E)*f(diag(9))
> c(LHS=LHS,RHS=RHS,diff=LHS-RHS) # LHS==RHS, according to Spivak's 4.6

|     | LHS   | RHS   | diff |
|-----|-------|-------|------|
|     | -9850953 | -9850953 | 0    |

Above we see agreement to within numerical precision.

5 Conclusions and further work

The stokes package furnishes functionality for working with tensors, alternating forms, and related concepts from the exterior calculus. Theorems including Stokes's generalized integral theorem were verified numerically. Further work might include working with Stokes’s theorem expressed in Clifford algebra formalism [1][4], following Klausen [6].

References

[1] Hankin, R. K. S. (2022a). Clifford algebra in R. https://arxiv.org/abs/2209.13659
[2] Hankin, R. K. S. (2022b). Disordered vectors in R: introducing the disordR package. https://arxiv.org/abs/2210.03856
[3] Hankin, R. K. S. (2022c). Sparse arrays in R: the spray package. https://arxiv.org/abs/2210.03856
[4] Hestenes, D. (1987). Clifford algebra to geometric calculus. Kluwer.
[5] Hubbard, J. J. and Hubbard, B. B. (2015). Vector calculus, linear algebra, and differential forms: a unified approach. Matrix Editions, fifth edition.
[6] Klausen, K. O. (2022). Visualizing Stokes’ theorem with geometric algebra. https://arxiv.org/abs/2206.07177
[7] R Core Team (2022). R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria.
[8] Spivak, M. (1965). Calculus on manifolds. Addison-Wesley.