

Package ‘stokes’

November 15, 2021

Type Package

Title The Exterior Calculus

Version 1.0-9

Depends spray

Suggests knitr,

Deriv,

testthat,

markdown,

rmarkdown,

emulator

Imports permutations (\geq 1.0-4), partitions, magrittr, methods, mathjaxr, disordR (\geq 0.0-8)

Maintainer Robin K. S. Hankin <hankin.robin@gmail.com>

Description Provides functionality for working with tensors, alternating tensors, wedge products, Stokes's theorem, and related concepts from the exterior calculus. Functionality for Grassman algebra is provided. The canonical reference would be: M. Spivak (1965, ISBN:0-8053-9021-9) ``Calculus on Manifolds''. Version stokes_1.0-8 is a reduced functionality placeholder pending upload of spray version 1.0-16 to CRAN.

License GPL-2

URL <https://github.com/RobinHankin/stokes>

BugReports <https://github.com/RobinHankin/stokes/issues>

RdMacros mathjaxr

R topics documented:

stokes-package	2
Alt	3
as.lform	5
coeffs	6
consolidate	6
contract	7
cross	8
hodge	9

inner	10
issmall	11
keep	11
kform	12
ktensor	14
Ops.kform	15
print.stokes	16
rform	17
scalar	18
symbolic	19
transform	19
volume	21
wedge	22
zap	22
zero	23

Index	24
--------------	-----------

stokes-package	<i>The Exterior Calculus</i>
----------------	------------------------------

Description

stokes version 1.0-9 is a reduced functionality placeholder pending upload of a newer spray package to CRAN which uses disordR discipline.

Provides functionality for working with tensors, alternating tensors, wedge products, Stokes's theorem, and related concepts from the exterior calculus. Functionality for Grassman algebra is provided. The canonical reference would be: M. Spivak (1965, ISBN:0-8053-9021-9) "Calculus on Manifolds". Version stokes_1.0-8 is a reduced functionality placeholder pending upload of spray version 1.0-16 to CRAN.

Details

The DESCRIPTION file:

```

Package:      stokes
Type:         Package
Title:        The Exterior Calculus
Version:      1.0-9
Depends:      spray
Suggests:    knitr, Deriv, testthat, markdown, rmarkdown, emulator
Imports:      permutations (>= 1.0-4), partitions, magrittr, methods, mathjaxr, disordR (>= 0.0-8)
Authors@R:   person( given=c("Robin", "K. S."), family="Hankin", role = c("aut", "cre"), email="hankin.robin@gmail.com)
Maintainer:  Robin K. S. Hankin <hankin.robin@gmail.com>
Description:  Provides functionality for working with tensors, alternating tensors, wedge products, Stokes's theorem, and
License:      GPL-2
URL:         https://github.com/RobinHankin/stokes
BugReports:  https://github.com/RobinHankin/stokes/issues
RdMacros:    mathjaxr
Author:      Robin K. S. Hankin [aut, cre] (<https://orcid.org/0000-0001-5982-0415>)

```

Index of help topics:

Alt	Alternating multilinear forms
Ops.kform	Arithmetic Ops Group Methods for 'kform' and 'ktensor' objects
as.1form	Coerce vectors to 1-forms
coeffs	Extract and manipulate coefficients
consolidate	Various low-level helper functions
contract	Contractions of k-forms
cross	Cross products of k-tensors
hodge	Hodge star operator
inner	Inner product operator
issmall	Is a form zero to within numerical precision?
keep	Keep or drop variables
kform	k-forms
ktensor	k-tensors
print.stokes	Print methods for k-tensors and k-forms
rform	Random kforms and ktensors
scalar	Lose attributes
stokes-package	The Exterior Calculus
symbolic	Symbolic form
transform	Linear transforms of k-forms
volume	The volume element
wedge	Wedge products
zap	Zap small values in k-forms and k-tensors
zeroform	Zero tensors and zero forms

Generally in the package, arguments that are k -forms are denoted K , k -tensors by U , and spray objects by S . Multilinear maps (which may be either k -forms or k -tensors) are denoted by M .

Author(s)

NA

Maintainer: Robin K. S. Hankin <hankin.robin@gmail.com>

References

- J. H. Hubbard and B. B. Hubbard 2015. *Vector calculus, linear algebra and differential forms: a unified approach*. Ithaca, NY.
- M. Spivak 1971. *Calculus on manifolds*, Addison-Wesley.

Alt

Alternating multilinear forms

Description

stokes version 1.0-9 is a reduced functionality placeholder package pending upload of a newer spray package to CRAN which uses `disordR` discipline.

Converts a k -tensor to alternating form

Usage

`Alt(S, give_kform)`

Arguments

<code>S</code>	A multilinear form, an object of class <code>ktensor</code>
<code>give_kform</code>	Boolean, with default <code>FALSE</code> meaning to return an alternating k -tensor [that is, an object of class <code>ktensor</code> that happens to be alternating] and <code>TRUE</code> meaning to return a k -form [that is, an object of class <code>kform</code>]

Details

Given a k -tensor T , we have

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

Thus for example if $k = 3$:

$$\text{Alt}(T)(v_1, v_2, v_3) = \frac{1}{6} \begin{pmatrix} +T(v_1, v_2, v_3) & -T(v_1, v_3, v_2) \\ -T(v_2, v_1, v_3) & +T(v_2, v_3, v_1) \\ +T(v_3, v_1, v_2) & -T(v_3, v_2, v_1) \end{pmatrix}$$

and it is reasonably easy to see that $\text{Alt}(T)$ is alternating, in the sense that

$$\text{Alt}(T)(v_1, \dots, v_i, \dots, v_j, \dots, v_k) = -\text{Alt}(T)(v_1, \dots, v_j, \dots, v_i, \dots, v_k)$$

Function `Alt()` is intended to take and return an object of class `ktensor`; but if given a `kform` object, it just returns its argument unchanged.

A short vignette is provided with the package: type `vignette("Alt")` at the commandline.

Value

Returns an alternating k -tensor. To work with k -forms, which are a much more efficient representation of alternating tensors, use `as.kform()`.

Author(s)

Robin K. S. Hankin

See Also

[kform](#)

as.1form

Coerce vectors to 1-forms

Description

stokes version 1.0-9 is a reduced functionality placeholder package pending upload of a newer spray package to CRAN which uses disordR discipline.

Given a vector, return the corresponding 1-form; the exterior derivative of a 0-form (that is, a scalar function). Function `grad()` is a synonym.

Usage

```
as.1form(v)
grad(v)
```

Arguments

`v` A vector with element i being $\partial f / \partial x_i$

Details

The exterior derivative of a k -form ϕ is a $(k + 1)$ -form $\mathbf{d}\phi$ given by

$$\mathbf{d}\phi(P_{\mathbf{x}}(\mathbf{v}_i, \dots, \mathbf{v}_{k+1})) = \lim_{h \rightarrow 0} \frac{1}{h^{k+1}} \int_{\partial P_{\mathbf{x}}(h\mathbf{v}_1, \dots, h\mathbf{v}_{k+1})} \phi$$

We can use the facts that

$$\mathbf{d}(f dx_{i_1} \wedge \dots \wedge dx_{i_k}) = \mathbf{d}f \wedge dx_{i_1} \wedge \dots \wedge dx_{i_k}$$

and

$$\mathbf{d}f = \sum_{j=1}^n (D_j f) dx_j$$

to calculate differentials of general k -forms. Specifically, if

$$\phi = \sum_{1 \leq i_1 < \dots < i_k \leq n} a_{i_1 \dots i_k} dx_{i_1} \wedge \dots \wedge dx_{i_k}$$

then

$$\mathbf{d}\phi = \sum_{1 \leq i_1 < \dots < i_k \leq n} \left[\sum_{j=1}^n D_j a_{i_1 \dots i_k} dx_j \right] \wedge dx_{i_1} \wedge \dots \wedge dx_{i_k}.$$

The entry in square brackets is given by `grad()`. See the examples for appropriate R idiom.

Value

A one-form

Author(s)

Robin K. S. Hankin

See Also[kform](#)

`coeffs`*Extract and manipulate coefficients*

Description

stokes Version 1.0-9 is a reduced functionality placeholder pending upload of a newer spray package to CRAN which uses disordR discipline.

Extract and manipulate coefficients of ktensor and kform objects

Arguments

<code>S</code>	Object to be tested for being a spray
<code>value</code>	In the assignment operator <code>coeffs<-()</code> , a disord object so that <code>coeffs(S) <-x</code> works as expected

Details

To see the coefficients of a kform or ktensor object, use `coeffs()`, which returns a disord object (this is actually `spray::coeffs()`). Replacement methods are specific to the **stokes** package but also use disord discipline.

Author(s)

Robin K. S. Hankin

`consolidate`*Various low-level helper functions*

Description

Various low-level helper functions used in `Alt()` and `kform()`

Usage

```
consolidate(S)
kill_trivial_rows(S)
include_perms(S)
kform_to_ktensor(S)
```

Arguments

<code>S</code>	Object of class spray
----------------	-----------------------

Details

Low-level helper functions.

- Function `consolidate()` takes a spray object, and combines any rows that are identical up to a permutation, respecting the sign of the permutation
- Function `kill_trivial_rows()` takes a spray object and deletes any rows with a repeated entry (which have k -forms identically zero)
- Function `include_perms()` replaces each row of a spray object with all its permutations, respecting the sign of the permutation
- Function `ktensor_to_kform()` coerces a k -form to a k -tensor

Value

The functions documented here all return a spray object.

Author(s)

Robin K. S. Hankin

See Also

[ktensor](#), [kform](#), [Alt](#)

contract

Contractions of k -forms

Description

A contraction is a natural linear map from k -forms to $k - 1$ -forms.

Usage

```
contract(K, v, lose=TRUE)
contract_elementary(o, v)
```

Arguments

<code>K</code>	A k -form
<code>o</code>	Integer-valued vector corresponding to one row of an index matrix
<code>lose</code>	Boolean, with default TRUE meaning to coerce a 0-form to a scalar and FALSE meaning to return the formal 0-form
<code>v</code>	A vector; in function <code>contract()</code> , if a matrix, interpret each column as a vector to contract with

Details

Given a k -form ϕ and a vector \mathbf{v} , the *contraction* $\phi_{\mathbf{v}}$ of ϕ and \mathbf{v} is a $k - 1$ -form with

$$\phi_{\mathbf{v}}(\mathbf{v}^1, \dots, \mathbf{v}^{k-1}) = \phi(\mathbf{v}, \mathbf{v}^1, \dots, \mathbf{v}^{k-1})$$

if $k > 1$; we specify $\phi_{\mathbf{v}} = \phi(\mathbf{v})$ if $k = 1$.

Function `contract_elementary()` is a low-level helper function that translates elementary k -forms with coefficient 1 (in the form of an integer vector corresponding to one row of an index matrix) into its contraction with \mathbf{v} .

Value

Returns an object of class `kform`.

Author(s)

Robin K. S. Hankin

References

Steven H. Weintraub 2014. “Differential forms: theory and practice”, Elsevier (contractions defined in Definition 2.2.23 in chapter 2, page 77).

See Also

[wedge,lose](#)

cross

Cross products of k -tensors

Description

stokes Version 1.0-9 is a reduced functionality placeholder pending upload of a newer spray package to CRAN which uses `disordR` discipline.

Cross products of k -tensors

Usage

```
cross(U, ...)
cross2(U1, U2)
```

Arguments

<code>U, U1, U2</code>	Object of class <code>ktensor</code>
<code>...</code>	Further arguments, currently ignored

Details

Given a k -tensor S and an l -tensor T , we can form the cross product $S \otimes T$, defined as

$$S \otimes T(v_1, \dots, v_k, v_{k+1}, \dots, v_{k+l}) = S(v_1, \dots, v_k) \cdot T(v_{k+1}, \dots, v_{k+l}).$$

Package idiom for this includes `cross(S,T)` and `S %% T`; note that the cross product is not commutative. Function `cross()` can take any number of arguments (the result is well-defined because the cross product is associative); it uses `cross2()` as a low-level helper function.

Value

The functions documented here all return a spray object.

Note

The binary form `%%` uses uppercase `X` to avoid clashing with `%%` which is the Kronecker product in base `R`.

Author(s)

Robin K. S. Hankin

References

Spivak 1961

See Also

[ktensor](#)

hodge

Hodge star operator

Description

stokes version 1.0-9 is a reduced functionality placeholder pending upload of a newer spray package to CRAN which uses `disordR` discipline.

Given a k -form, return its Hodge dual

Usage

```
hodge(K, n=max(index(K)), g=rep(1,n), lose=TRUE)
```

Arguments

<code>K</code>	Object of class <code>kform</code>
<code>n</code>	Dimensionality of space, defaulting to the largest element of the index
<code>g</code>	Diagonal of the metric tensor, defaulting to the standard metric
<code>lose</code>	Boolean, with default <code>TRUE</code> meaning to coerce to a scalar if appropriate

Value

Given a k -form, in an n -dimensional space, returns a $(n - k)$ -form.

Author(s)

Robin K. S. Hankin

See Also

[wedge](#)

inner

Inner product operator

Description

The inner product

Usage

`inner(M)`

Arguments

`M` square matrix

Details

The inner product of two vectors \mathbf{x} and \mathbf{y} is usually written $\langle \mathbf{x}, \mathbf{y} \rangle$ or $\mathbf{x} \cdot \mathbf{y}$, but the most general form would be $\mathbf{x}^T M \mathbf{y}$ where M is a matrix. Noting that inner products are multilinear, that is $\langle \mathbf{x}, a\mathbf{y} + b\mathbf{z} \rangle = a \langle \mathbf{x}, \mathbf{y} \rangle + b \langle \mathbf{x}, \mathbf{z} \rangle$ and $\langle a\mathbf{x} + b\mathbf{y}, \mathbf{z} \rangle = a \langle \mathbf{x}, \mathbf{z} \rangle + b \langle \mathbf{y}, \mathbf{z} \rangle$, we see that the inner product is indeed a multilinear map, that is, a tensor.

Given a square matrix M , function `inner(M)` returns the 2-form that maps \mathbf{x}, \mathbf{y} to $\mathbf{x}^T M \mathbf{y}$.

A short vignette is provided with the package: `type vignette("inner")` at the commandline.

Value

Returns a k -tensor, an inner product

Author(s)

Robin K. S. Hankin

See Also

[kform](#)

issmall	<i>Is a form zero to within numerical precision?</i>
---------	--

Description

stokes Version 1.0-9 is a reduced functionality placeholder pending upload of a newer spray package to CRAN which uses disordR discipline.

Given a k -form, return TRUE if it is “small”

Usage

```
issmall(M, tol=1e-8)
```

Arguments

M	Object of class kform or ktensor
tol	Small tolerance, defaulting to 1e-8

Value

Returns a logical

Author(s)

Robin K. S. Hankin

keep	<i>Keep or drop variables</i>
------	-------------------------------

Description

stokes version 1.0-9 is a reduced functionality placeholder pending upload of a newer spray package to CRAN which uses disordR discipline.

Keep or drop variables

Usage

```
keep(K, yes)
discard(K, no)
```

Arguments

K	Object of class kform
yes, no	Specification of dimensions to either keep (yes) or discard (no), coerced to a free object

Details

Function `keep(omega, yes)` keeps the terms specified and `discard(omega, no)` discards the terms specified. It is not clear to me what these functions mean from a mathematical perspective.

Value

The functions documented here all return a kform object.

Author(s)

Robin K. S. Hankin

See Also

[lose](#)

kform	<i>k-forms</i>
-------	----------------

Description

Functionality for dealing with k -forms

Usage

```
kform(S)
as.kform(M, coeffs, lose=TRUE)
kform_basis(n, k)
kform_general(W, k, coeffs, lose=TRUE)
is.kform(x)
## S3 method for class 'kform'
as.function(x, ...)
```

Arguments

n	Dimension of the vector space $V = R^n$
k	A k -form maps V^k to R
W	Integer vector of dimensions
M, coeffs	Index matrix and coefficients for a k -form
S	Object of class spray
lose	Boolean, with default TRUE meaning to coerce a 0-form to a scalar and FALSE meaning to return the formal 0-form
x	Object of class kform
...	Further arguments, currently ignored

Details

A k -form is an alternating k -tensor. In the **stokes** package, k -forms are represented as sparse arrays (spray objects), but with a class of c("kform", "spray"). The constructor function kform() takes a spray object and returns a kform object: it ensures that rows of the index matrix are strictly nonnegative integers, have no repeated entries, and are strictly increasing. Function as.kform() is more user-friendly.

- kform() is the constructor function. It takes a spray object and returns a kform.

- `as.kform()` also returns a `kform` but is a bit more user-friendly than `kform()`.
- `kform_basis()` is a low-level helper function that returns a matrix whose rows constitute a basis for the vector space $\Lambda^k(R^n)$ of k -forms.
- `kform_general()` returns a `kform` object with terms that span the space of alternating tensors.
- `is.kform()` returns `TRUE` if its argument is a `kform` object.

Recall that a k -tensor is a multilinear map from V^k to the reals, where $V = R^n$ is a vector space. A multilinear k -tensor T is *alternating* if it satisfies

$$T(v_1, \dots, v_i, \dots, v_j, \dots, v_k) = T(v_1, \dots, v_j, \dots, v_i, \dots, v_k)$$

In the package, an object of class `kform` is an efficient representation of an alternating tensor.

Function `kform_basis()` is a low-level helper function that returns a matrix whose rows constitute a basis for the vector space $\Lambda^k(R^n)$ of k -forms:

$$\phi = \sum_{1 \leq i_1 < \dots < i_k \leq n} a_{i_1 \dots i_k} dx_{i_1} \wedge \dots \wedge dx_{i_k}$$

and indeed we have:

$$a_{i_1 \dots i_k} = \phi(\mathbf{e}_{i_1}, \dots, \mathbf{e}_{i_k})$$

where $\mathbf{e}_j, 1 \leq j \leq k$ is a basis for V .

Value

All functions documented here return a `kform` object except `as.function.kform()`, which returns a function, and `is.kform()`, which returns a Boolean.

Note

Hubbard and Hubbard use the term “ k -form”, but Spivak does not.

Author(s)

Robin K. S. Hankin

References

Hubbard and Hubbard; Spivak

See Also

[ktensor, lose](#)

ktensor	<i>k</i> -tensors
---------	-------------------

Description

Functionality for *k*-tensors

Usage

```
ktensor(S)
as.ktensor(M, coeffs)
is.ktensor(x)
## S3 method for class 'ktensor'
as.function(x, ...)
```

Arguments

M, coeffs	Matrix of indices and coefficients, as in <code>spray(M, coeffs)</code>
S	Object of class <code>spray</code>
x	Object of class <code>ktensor</code>
...	Further arguments, currently ignored

Details

A *k*-tensor object S is a map from V^k to the reals R , where V is a vector space (here R^n) that satisfies multilinearity:

$$S(v_1, \dots, av_i, \dots, v_k) = a \cdot S(v_1, \dots, v_i, \dots, v_k)$$

and

$$S(v_1, \dots, v_i + v_i', \dots, v_k) = S(v_1, \dots, v_i, \dots, v_k) + S(v_1, \dots, v_i', \dots, v_k).$$

Note that this is *not* equivalent to linearity over V^{nk} (see examples).

In the **stokes** package, *k*-tensors are represented as sparse arrays (`spray` objects), but with a class of `c("ktensor", "spray")`. This is a natural and efficient representation for tensors that takes advantage of sparsity using **spray** package features.

Function `as.ktensor()` will coerce a *k*-form to a *k*-tensor via `kform_to_ktensor()`.

Value

All functions documented here return a `ktensor` object except `as.function.ktensor()`, which returns a function.

Author(s)

Robin K. S. Hankin

References

Spivak 1961

See Also

[cross,kform,wedge](#)

 Ops.kform

Arithmetic Ops Group Methods for kform and ktensor objects

Description

Allows arithmetic operators to be used for k -forms and k -tensors such as addition, multiplication, etc, where defined.

Usage

```
## S3 method for class 'kform'
Ops(e1, e2 = NULL)
## S3 method for class 'ktensor'
Ops(e1, e2 = NULL)
```

Arguments

`e1, e2` Objects of class `kform` or `ktensor`

Details

The functions `Ops.kform()` and `Ops.ktensor()` pass unary and binary arithmetic operators (“+”, “-”, “*”, and “/”) to the appropriate specialist function by coercing to spray objects.

For wedge products of k -forms, use `wedge()` or `%^%`; and for cross products of k -tensors, use `cross()` or `%X%`.

Value

All functions documented here return an object of class `kform` or `ktensor`.

Note

A plain asterisk, “*”, given two `kforms` or two `ktensors`, will return the wedge product or the cross product respectively, on the grounds that the idiom has only one natural interpretation. But its use is discouraged.

Author(s)

Robin K. S. Hankin

print.stokes	<i>Print methods for k-tensors and k-forms</i>
--------------	--

Description

stokes Version 1.0-9 is a reduced functionality placeholder pending upload of a newer spray package to CRAN which uses disordR discipline.

Print methods for objects with options for printing in matrix form or multivariate polynomial form

Usage

```
## S3 method for class 'kform'
print(x, ...)
## S3 method for class 'ktensor'
print(x, ...)
```

Arguments

<code>x</code>	k -form or k -tensor
<code>...</code>	Further arguments (currently ignored)

Details

The print method is designed to tell the user that an object is a tensor or a k -form. It prints a message to this effect (with special dispensation for zero tensors), then calls the spray print method.

Value

Returns its argument invisibly.

Note

The print method asserts that its argument is a map from R^n to R , where n is the largest element in the index matrix. However, such a map naturally furnishes a map from R^m to R provided that $m \geq n$ via the natural projection from R^n to R^m . Formally this would be $(x_1, \dots, x_n) \mapsto (x_1, \dots, x_n, 0, \dots, 0) \in R^m$. In the case of the zero k -form or k -tensor, “ n ” is to be interpreted as “any $n \geq 0$ ”.

Author(s)

Robin K. S. Hankin

rform *Random kforms and ktensors*

Description

stokes version 1.0-9 is a reduced functionality placeholder package pending upload of a newer spray package to CRAN which uses disordR discipline.

Random k -form objects and k -tensors, intended as quick “get you going” examples

Usage

```
rform(terms=9,k=3,n=7,coeffs)
rtensor(terms=9,k=3,n=7,coeffs)
```

Arguments

terms	Number of distinct terms
k,n	A k -form maps V^k to R , where $V = R^n$
coeffs	The coefficients of the form; if missing use <code>seq_len(terms)</code>

Details

What you see is what you get, basically.

Note that argument `terms` is an upper bound, as the index matrix might contain repeats which are combined.

Value

All functions documented here return an object of class `kform` or `ktensor`.

Author(s)

Robin K. S. Hankin

Examples

```
1+1 # stokes_1.0-9 is a placeholder pending release of spray
```

scalar *Lose attributes*

Description

stokes version 1.0-9 is a reduced functionality placeholder pending upload of a newer spray package to CRAN which uses disordR discipline.

Scalars: 0-forms and 0-tensors

Usage

```
scalar(s,lose=FALSE)
is.scalar(M)
`0form`(s,lose=FALSE)
## S3 method for class 'kform'
lose(M)
## S3 method for class 'ktensor'
lose(M)
```

Arguments

s	A scalar value; a number
M	Object of class ktensor or kform
lose	In function scalar(), Boolean with TRUE meaning to return a normal scalar, and default FALSE meaning to return a formal 0-form or 0-tensor

Details

A k -tensor (including k -forms) maps k vectors to a scalar. If $k = 0$, then a 0-tensor maps no vectors to a scalar, that is, mapping nothing at all to a scalar, or what normal people would call a plain old scalar. Such forms are created by a couple of constructions in the package, specifically `scalar()`, `kform_general(1, 0)` and `contract()`. These functions take a `lose` argument that behaves much like the `drop` argument in base extraction.

Function `lose()` takes an object of class `ktensor` or `kform` and, if of arity zero, returns the coefficient.

Note that function `kform()` *always* returns a `kform` object, it never loses attributes.

A 0-form is not the same thing as a zero tensor. A 0-form maps V^0 to the reals; a scalar. A zero tensor maps V^k to zero.

Value

The functions documented here return an object of class `kform` or `ktensor`, except for `is.scalar()`, which returns a Boolean.

Author(s)

Robin K. S. Hankin

See Also

[zeroform](#), [lose](#)

symbolic	<i>Symbolic form</i>
----------	----------------------

Description

stokes version 1.0-9 is a reduced functionality placeholder pending upload of a newer spray package to CRAN which uses disordR discipline.

Prints k -tensor and k -form objects in symbolic form

Usage

```
as.symbolic(M, symbols=letters, d="")
```

Arguments

M	Object of class kform or ktensor; a map from V^k to R , where $V = R^n$
symbols	A character vector giving the names of the symbols
d	String specifying the appearance of the differential operator

Value

Returns a noquote character string.

Author(s)

Robin K. S. Hankin

transform	<i>Linear transforms of k-forms</i>
-----------	-------------------------------------

Description

Given a k -form, express it in terms of linear combinations of the dx_i

Usage

```
transform(K, M)
stretch(K, d)
```

Arguments

K	Object of class kform
M	Matrix of transformation
d	Numeric vector representing the diagonal elements of a diagonal matrix

Details

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

$$\omega = \sum_{i_1 < \dots < i_k} a_{i_1 \dots i_k} dx_{i_1} \wedge \dots \wedge dx_{i_k}$$

giving

$$\omega = \sum_{i_1 < \dots < i_k} \left[a_{i_1, \dots, i_k} \left(\sum_r M_{i_1 r} dy_r \right) \wedge \dots \wedge \left(\sum_r M_{i_k r} dy_r \right) \right].$$

The `transform()` function does all this but it is slow. 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` and a discussion in the `stokes` vignette.

Function `stretch()` carries out the same operation but for M a diagonal matrix. It is much faster than `transform()`.

Value

The functions documented here return an object of class `kform`.

Author(s)

Robin K. S. Hankin

References

S. H. Weintraub 2019. *Differential forms: theory and practice*. Elsevier. (Chapter 3)

See Also

[wedge](#)

volume	<i>The volume element</i>
--------	---------------------------

Description

stokes Version 1.0-9 is a reduced functionality placeholder pending upload of a newer spray package to CRAN which uses disordR discipline.

The volume element in n dimensions

Usage

```
volume(n)
is.volume(K)
```

Arguments

n	Dimension of the space
K	Object of class kform

Details

Spivak phrases it well (theorem 4.6, page 82):

If V has dimension n , it follows that $\Lambda^n(V)$ has dimension 1. Thus all alternating n -tensors on V are multiples of any non-zero one. Since the determinant is an example of such a member of $\Lambda^n(V)$ it is not surprising to find it in the following theorem:

Let v_1, \dots, v_n be a basis for V and let $\omega \in \Lambda^n(V)$. If $w_i = \sum_{j=1}^n a_{ij}v_j$ then

$$\omega(w_1, \dots, w_n) = \det(a_{ij}) \cdot \omega(v_1, \dots, v_n)$$

(see the examples for numerical verification of this).

Neither the zero k -form, nor scalars, are considered to be a volume element.

Value

Function `volume()` returns an object of class `kform`; function `is.volume()` returns a Boolean.

Author(s)

Robin K. S. Hankin

References

Spivak

See Also

[zeroform, as.1form](#)

 wedge

Wedge products

Description

Wedge products of k -forms

Usage

```
wedge2(K1,K2)
wedge(x, ...)
```

Arguments

$K1, K2, x, \dots$ k -forms

Details

Wedge product of k -forms.

Value

The functions documented here return an object of class `kform`.

Note

In general use, use `wedge()` or `%^%`. Function `wedge()` uses low-level helper function `wedge2()`, which takes only two arguments.

A short vignette is provided with the package: type `vignette("wedge")` at the commandline.

Author(s)

Robin K. S. Hankin

zap

Zap small values in k -forms and k -tensors

Description

stokes Version 1.0-9 is a reduced functionality placeholder pending upload of a newer spray package to CRAN which uses `disordR` discipline.

Equivalent to `zapsmall()`

Usage

```
zap(X)
## S3 method for class 'kform'
zap(X)
## S3 method for class 'ktensor'
zap(X)
```

Arguments

`X` Tensor or k -form to be zapped

Details

Given an object of class `ktensor` or `kform`, coefficients close to zero are ‘zapped’, i.e., replaced by ‘0’, using `base::zapsmall()`.

Note, `zap()` actually changes the numeric value, it is not just a print method.

Value

Returns an object of the same class

Author(s)

Robin K. S. Hankin

zero

Zero tensors and zero forms

Description

stokes version 1.0-9 is a reduced functionality placeholder package pending upload of a newer spray package to CRAN which uses `disordR` discipline.

Correct idiom for generating zero k -tensors and k -forms

Usage

```
zeroform(n)
zerotensor(n)
```

Arguments

`n` Arity of the k -form or k -tensor

Value

Returns an object of class `kform` or `ktensor`.

Note

Idiom such as `as.ktensor(rep(1,n),0)` and `as.kform(rep(1,5),0)` and indeed `as.kform(1:5,0)` is incorrect as the arity of the tensor is lost.

A 0-form is not the same thing as a zero tensor. A 0-form maps V^0 to the reals; a scalar. A zero tensor maps V^k to zero.

Author(s)

Robin K. S. Hankin

See Also

[scalar](#)

Index

- * **package**
 - stokes-package, 2
- * **symbolmath**
 - coeffs, 6
 - Ops.kform, 15
 - print.stokes, 16
- %X% (cross), 8
- %^% (wedge), 22
- \emptyset form (scalar), 18

- Alt, 3, 7
- as.1form, 5, 21
- as.function.kform (kform), 12
- as.function.ktensor (ktensor), 14
- as.kform (kform), 12
- as.ktensor (ktensor), 14
- as.symbolic (symbolic), 19

- coeff (coeffs), 6
- coeffs, 6
- coeffs,kform-method (coeffs), 6
- coeffs,ktensor-method (coeffs), 6
- coeffs.kform (coeffs), 6
- coeffs.ktensor (coeffs), 6
- coeffs<- (coeffs), 6
- coeffs<- ,kform-method (coeffs), 6
- coeffs<- ,ktensor-method (coeffs), 6
- coeffs<- .kform (coeffs), 6
- coeffs<- .ktensor (coeffs), 6
- consolidate, 6
- contract, 7
- contract_elementary (contract), 7
- cross, 8, 15
- cross2 (cross), 8

- discard (keep), 11
- drop (scalar), 18
- drop.free (keep), 11

- general_kform (kform), 12
- grad (as.1form), 5

- Hodge (hodge), 9
- hodge, 9

- include_perms (consolidate), 6
- inner, 10
- inner_product (inner), 10
- is.form (kform), 12
- is.kform (kform), 12
- is.ktensor (ktensor), 14
- is.scalar (scalar), 18
- is.tensor (ktensor), 14
- is.volume (volume), 21
- issmall, 11

- keep, 11
- kform, 4, 6, 7, 10, 12, 15
- kform_basis (kform), 12
- kform_general (kform), 12
- kform_to_ktensor (consolidate), 6
- kill_trivial_rows (consolidate), 6
- ktensor, 7, 9, 13, 14

- lose, 8, 12, 13, 18
- lose (scalar), 18
- lose_repeats (consolidate), 6

- Ops (Ops.kform), 15
- Ops.kform, 15

- print.kform (print.stokes), 16
- print.ktensor (print.stokes), 16
- print.stokes, 16
- pull-back (transform), 19
- pullback (transform), 19
- push-forward (transform), 19
- pushforward (transform), 19

- retain (keep), 11
- rform, 17
- rkform (rform), 17
- rktensor (rform), 17
- rtensor (rform), 17

- scalar, 18, 23
- star (hodge), 9
- stokes-package, 2
- stretch (transform), 19
- symbolic, 19

transform, [19](#)

value<- (coeffs), [6](#)

volume, [21](#)

wedge, [8](#), [10](#), [15](#), [20](#), [22](#)

wedge2 (wedge), [22](#)

zap, [22](#)

zapsmall (zap), [22](#)

zaptiny (zap), [22](#)

zero, [23](#)

zeroform, [18](#), [21](#)

zeroform (zero), [23](#)

zerotensor (zero), [23](#)