16.45 ORTHOVEC: Manipulation of scalars and vectors

ORTHOVEC is a collection of REDUCE procedures and operations which provide a simple-to-use environment for the manipulation of scalars and vectors. Operations include addition, subtraction, dot and cross products, division, modulus, div, grad, curl, laplacian, differentiation, integration, and Taylor expansion.

Author: James W. Eastwood.

Version 2 is summarized in [2]. It differs from the original ([1]) in revised notation and extended capabilities.

16.45.1 Introduction

The revised version of ORTHOVEC[2] is, like the original[1], a collection of REDUCE procedures and operators designed to simplify the machine aided manipulation of vectors and vector expansions frequently met in many areas of applied mathematics. The revisions have been introduced for two reasons: firstly, to add extra capabilities missing from the original and secondly, to tidy up input and output to make the package easier to use.

The changes from Version 1 include:

1. merging of scalar and vector unary and binary operators, $+, -, *, /$
2. extensions of the definitions of division and exponentiation to vectors
3. new vector dependency procedures
4. application of l'Hôpital's rule in limits and Taylor expansions
5. a new component selector operator
6. algebraic mode output of LISP vector components

The LISP vector primitives are again used to store vectors, although with the introduction of LIST types in algebraic mode in REDUCE 3.4, the implementation may have been more simply achieved using lists to store vector components.

The philosophy used in Version 2 follows that used in the original: namely, algebraic mode is used wherever possible. The view is taken that some computational inefficiencies are acceptable if it allows coding to be intelligible to (and thence adaptable by) users other than LISP experts familiar with the internal workings of REDUCE.

Procedures and operators in ORTHOVEC fall into the five classes: initialisation, input-output, algebraic operations, differential operations and integral operations.

Definitions are given in the following sections, and a summary of the procedure names and their meanings are give in Table 1. The final section discusses test examples.

16.45.2 Initialisation

The procedure VSTART initialises ORTHOVEC. It may be called after ORTHOVEC has been INputted (or LOADed if a fast load version has been made) to reset coordinates. VSTART provides a menu of standard coordinate systems:-

1. cartesian (x, y, z) = `(x, y, z)`
2. cylindrical (r, θ, z) = `(r, th, z)`
3. spherical (r, θ, ϕ) = `(r, th, ph)`
4. general (u_1, u_2, u_3) = `(u1, u2, u3)`
5. others

which the user selects by number. Selecting options (1)-(4) automatically sets up the coordinates and scale factors. Selection option (5) shows the user how to select another coordinate system. If VSTART is not called, then the default cartesian coordinates are used. ORTHOVEC may be re-initialised to a new coordinate system at any time during a given REDUCE session by typing

```
VSTART $.
```

16.45.3 Input-Output

ORTHOVEC assumes all quantities are either scalars or 3 component vectors. To define a vector a with components (c_1, c_2, c_3) use the procedure SVEC as follows

```
a := svec(c1, c2, c3);
```

The standard REDUCE output for vectors when using the terminator ";" is to list the three components inside square brackets $[\cdots]$, with each component in prefix form. A replacement for the standard REDUCE procedure MAPRIN is included in the package to change the output of LISP vector components to algebraic notation. The procedure VOUT (which returns the value of its argument) can be used to give labelled output of components in algebraic form: e.g.,

```
b := svec (sin(x)**2, y**2, z)$
vout(b)$
```

The operator _ can be used to select a particular component (1, 2 or 3) for output e.g.

```
b_1 ;
```

16.45.4 Algebraic Operations

Six infix operators, sum, difference, quotient, times, exponentiation and cross product, and four prefix operators, plus, minus, reciprocal and modulus are defined in ORTHOVEC. These operators can take suitable combinations of scalar and vector arguments, and in the case of scalar arguments reduce to the usual definitions of $+, -, *, /$, etc.

The operators are represented by symbols

```
+, -, /, *, ^, ><
```

The composite >< is an attempt to represent the cross product symbol × in ASCII characters. If we let $\mathbf{v}$ be a vector and s be a scalar, then valid combinations of arguments of the procedures and operators and the type of the result are as summarised below. The notation used is
result :=procedure(left argument, right argument) or
result :=(left operand) operator (right operand) .

Vector Addition

v	:=	VECTORPLUS(**v**)	or	**v**	:=	+ **v**
s	:=	VECTORPLUS(s)	or	s	:=	+ s
v	:=	VECTORADD(**v**,**v**)	or	**v**	:=	**v** + **v**
s	:=	VECTORADD(s,s)	or	s	:=	s + s

Vector Subtraction

v	:=	VECTORMINUS(**v**)	or	**v**	:=	- **v**
s	:=	VECTORMINUS(s)	or	s	:=	- s
v	:=	VECTORDIFFERENCE(**v**,**v**)	or	**v**	:=	**v** - **v**
s	:=	VECTORDIFFERENCE(s,s)	or	s	:=	s - s

Vector Division

v	:=	VECTORRECIP(**v**)	or	**v**	:=	/ **v**
s	:=	VECTORRECIP(s)	or	s	:=	/ s
v	:=	VECTORQUOTIENT(**v**,**v**)	or	**v**	:=	**v** / **v**
v	:=	VECTORQUOTIENT(**v**, s)	or	**v**	:=	**v** / s
v	:=	VECTORQUOTIENT(s ,**v**)	or	**v**	:=	s / **v**
s	:=	VECTORQUOTIENT(s,s)	or	s	:=	s / s

Vector Multiplication

v	:=	VECTORTIMES(s ,**v**)	or	**v**	:=	s * **v**
v	:=	VECTORTIMES(**v**, s)	or	**v**	:=	**v** * s
s	:=	VECTORTIMES(**v**,**v**)	or	s	:=	**v** * **v**
s	:=	VECTORTIMES(s , s)	or	s	:=	s * s

Vector Cross Product

v	:=	VECTORCROSS(**v**,**v**)	or	**v**	:=	**v** × **v**

Vector Exponentiation

s	:=	VECTOREXPT (**v**, s)	or	s	:=	**v** ^ s
s	:=	VECTOREXPT (s , s)	or	s	:=	s ^ s

Vector Modulus

s	:=	VMOD (s)
s	:=	VMOD (**v**)

All other combinations of operands for these operators lead to error messages being issued. The first two instances of vector multiplication are scalar multiplication of vectors, the third is the product of two scalars and the last is the inner (dot) product. The unary operators +, −, / can take either scalar or vector arguments and return results of the same type as their arguments. VMOD returns a scalar.

In compound expressions, parentheses may be used to specify the order of combi-

s	:=	div ($\mathbf{v}$)
$\mathbf{v}$	:=	grad(s)
$\mathbf{v}$	:=	curl($\mathbf{v}$)
$\mathbf{v}$	:=	delsq($\mathbf{v}$)
s	:=	delsq(s)
$\mathbf{v}$	:=	$\mathbf{v}$ dotgrad $\mathbf{v}$
s	:=	$\mathbf{v}$ dotgrad s

Table 16.2: ORTHOVEC valid combinations of operator and argument

nation. If parentheses are omitted the ordering of the operators, in increasing order of precedence is

```
+ | - | dotgrad | * | >< | ^ | _
```

and these are placed in the precedence list defined in REDUCE after <. The differential operator DOTGRAD is defined in the following section, and the component selector _ was introduced in section 3.

Vector divisions are defined as follows: If $\mathbf{a}$ and $\mathbf{b}$ are vectors and c is a scalar, then

$$\begin{aligned} \mathbf{a}/\mathbf{b} &= \frac{\mathbf{a}\cdot\mathbf{b}}{\mid \mathbf{b} \mid^2} \\ c/\mathbf{a} &= \frac{c\mathbf{a}}{\mid \mathbf{a} \mid^2} \end{aligned}$$

Both scalar multiplication and dot products are given by the same symbol, braces are advisable to ensure the correct precedences in expressions such as $(\mathbf{a}\cdot\mathbf{b})(\mathbf{c}\cdot\mathbf{d})$.

Vector exponentiation is defined as the power of the modulus:
$\mathbf{a}^n \equiv \mathrm{VMOD}(a)^n = \mid \mathbf{a} \mid^n$

16.45.5 Differential Operations

Differential operators provided are div, grad, curl, delsq, and dotgrad. All but the last of these are prefix operators having a single vector or scalar argument as appropriate. Valid combinations of operator and argument, and the type of the result are shown in table 16.2.

All other combinations of operator and argument type cause error messages to be issued. The differential operators have their usual meanings [3]. The coordinate system used by these operators is set by invoking VSTART (cf. Sec. 16.45.2). The names `h1`, `h2` and `h3` are reserved for the scale factors, and `u1`, `u2` and `u3` are used for the coordinates.

VEX	VX	VPT	VORDER
v	**v**	**v**	**v**
v	**v**	**v**	s
v	s	s	s
s	**v**	**v**	**v**
s	**v**	**v**	s
s	s	s	s

Table 16.3: ORTHOVEC valid combination of argument types.

A vector extension, VDF, of the REDUCE procedure DF allows the differentiation of a vector (scalar) with respect to a scalar to be performed. Allowed forms are VDF(**v**, s) → **v** and VDF(s, s) → s , where, for example

$$\texttt{vdf(B,x)} \equiv \frac{\partial \mathbf{B}}{\partial x}$$

The standard REDUCE procedures DEPEND and NODEPEND have been redefined to allow dependences of vectors to be compactly defined. For example

```
a := svec(a1,a2,a3)$;
depend a,x,y;
```

causes all three components `a1`,`a2` and `a3` of `a` to be treated as functions of `x` and `y`. Individual component dependences can still be defined if desired.

```
depend a3,z;
```

The procedure VTAYLOR gives truncated Taylor series expansions of scalar or vector functions:-

```
vtaylor(vex,vx,vpt,vorder);
```

returns the series expansion of the expression VEX with respect to variable VX about point VPT to order VORDER. Valid combinations of argument types are shown in table 16.3.

Any other combinations cause error messages to be issued. Elements of VORDER must be non-negative integers, otherwise error messages are issued. If scalar VORDER is given for a vector expansion, expansions in each component are truncated at the same order, VORDER.

The new version of Taylor expansion applies l'Hôpital's rule in evaluating coefficients, so handle cases such as $\sin(x)/(x)$, etc. which the original version of ORTHOVEC could not. The procedure used for this is LIMIT, which can be used directly to find the limit of a scalar function `ex` of variable `x` at point `pt`:-

```
ans := limit(ex,x,pt);
```

16.45.6 Integral Operations

Definite and indefinite vector, volume and scalar line integration procedures are included in ORTHOVEC. They are defined as follows:

$$\begin{aligned}
\mathrm{VINT}(\mathbf{v},x) &= \int \mathbf{v}(x)dx \\
\mathrm{DVINT}(\mathbf{v},x,a,b) &= \int_a^b \mathbf{v}(x)dx \\
\mathrm{VOLINT}(\mathbf{v}) &= \int \mathbf{v}h_1h_2h_3du_1du_2du_3 \\
\mathrm{DVOLINT}(\mathbf{v},\mathbf{l},\mathbf{u},n) &= \int_{\mathbf{l}}^{\mathbf{u}} \mathbf{v}h_1h_2h_3du_1du_2du_3 \\
\mathrm{LINEINT}(\mathbf{v},\omega,t) &= \int \mathbf{v}\cdot\mathbf{dr} \equiv \int v_ih_i\frac{\partial\omega_i}{\partial t}dt \\
\mathrm{DLINEINT}(\mathbf{v},\omega t,a,b) &= \int_a^b v_ih_i\frac{\partial\omega_i}{\partial t}dt
\end{aligned}$$

In the vector and volume integrals, $\mathbf{v}$ are vector or scalar, a, b, x and n are scalar. Vectors $\mathbf{l}$ and $\mathbf{u}$ contain expressions for lower and upper bounds to the integrals. The integer index n defines the order in which the integrals over u_1, u_2 and u_3 are performed in order to allow for functional dependencies in the integral bounds:

n	order
1	$u_1\ u_2\ u_3$
2	$u_3\ u_1\ u_2$
3	$u_2\ u_3\ u_1$
4	$u_1\ u_3\ u_2$
5	$u_2\ u_1\ u_3$
otherwise	$u_3\ u_2\ u_1$

The vector ω in the line integral's arguments contain explicit paramterisation of the coordinates u_1, u_2, u_3 of the line $\mathbf{u}(t)$ along which the integral is taken.

Procedures		Description
VSTART		select coordinate system
SVEC		set up a vector
VOUT		output a vector
VECTORCOMPONENT	_	extract a vector component (1-3)
VECTORADD	+	add two vectors or scalars
VECTORPLUS	+	unary vector or scalar plus
VECTORMINUS	-	unary vector or scalar minus
VECTORDIFFERENCE	-	subtract two vectors or scalars
VECTORQUOTIENT	/	vector divided by scalar
VECTORRECIP	/	unary vector or scalar division (reciprocal)
VECTORTIMES	*	multiply vector or scalar by vector/scalar
VECTORCROSS	><	cross product of two vectors
VECTOREXPT	^	exponentiate vector modulus or scalar
VMOD		length of vector or scalar

Table 16.4: Procedures names and operators used in ORTHOVEC (part 1)

16.45.7 Test Cases

To use the REDUCE source version of ORTHOVEC, initiate a REDUCE session and then IN the file *orthovec.red* containing ORTHOVEC. However, it is recommended that for efficiency a compiled fast loading version be made and LOADed when required (see Sec. 18 of the REDUCE manual). If coordinate dependent differential and integral operators other than cartesian are needed, then VSTART must be used to reset coordinates and scale factors.

Six simple examples are given in the Test Run Output file *orthovectest.log* to illustrate the working of ORTHOVEC. The input lines were taken from the file *orthovectest.red* (the Test Run Input), but could equally well be typed in at the Terminal.

Example 33

Show that

$$(\mathbf{a} \times \mathbf{b}) \cdot (\mathbf{c} \times \mathbf{d}) - (\mathbf{a} \cdot \mathbf{c})(\mathbf{b} \cdot \mathbf{d}) + (\mathbf{a} \cdot \mathbf{d})(\mathbf{b} \cdot \mathbf{c}) \equiv 0$$

Example 34

Procedures	Description
DIV	divergence of vector
GRAD	gradient of scalar
CURL	curl of vector
DELSQ	laplacian of scalar or vector
DOTGRAD	(vector).grad(scalar or vector)
VTAYLOR	vector or scalar Taylor series of vector or scalar
VPTAYLOR	vector or scalar Taylor series of scalar
TAYLOR	scalar Taylor series of scalar
LIMIT	limit of quotient using l'Hôpital's rule
VINT	vector integral
DVINT	definite vector integral
VOLINT	volume integral
DVOLINT	definite volume integral
LINEINT	line integral
DLINEINT	definite line integral
MAPRIN	vector extension of REDUCE MAPRIN
DEPEND	vector extension of REDUCE DEPEND
NODEPEND	vector extension of REDUCE NODEPEND

Table 16.5: Procedures names and operators used in ORTHOVEC (part 2)

Write the equation of motion

$$\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v} + \nabla p - curl(\mathbf{B}) \times \mathbf{B}$$

in cylindrical coordinates.

Example 35

Taylor expand

- $\sin(x)\cos(y) + e^z$ about the point $(0, 0, 0)$ to third order in x, fourth order in y and fifth order in z.
- $\sin(x)/x$ about x to fifth order.
- $\mathbf{v}$ about $\mathbf{x} = (x, y, z)$ to fifth order, where $\mathbf{v} = (x/\sin(x), (e^y - 1)/y, (1 + z)^{10})$.

Example 36

Obtain the second component of the equation of motion in example 34, and the first component of the final vector Taylor series in example 35.

Example 37

Evaluate the line integral

$$\int_{\mathbf{r}_1}^{\mathbf{r}_2} \mathbf{A} \cdot d\mathbf{r}$$

from point $\mathbf{r}_1 = (1, 1, 1)$ to point $\mathbf{r}_2 = (2, 4, 8)$ along the path $(x, y, z) = (s, s^2, s^3)$ where

$$\mathbf{A} = (3x^2 + 5y)\mathbf{i} - 12xy\mathbf{j} + 2xyz^2\mathbf{k}$$

and $(\mathbf{i}, \mathbf{j}, \mathbf{k})$ are unit vectors in the (x, y, z) directions.

Example 38

Find the volume V common to the intersecting cylinders $x^2 + y^2 = r^2$ and $x^2 + z^2 = r^2$ i.e. evaluate

$$V = 8 \int_0^r dx \int_0^{ub} dy \int_0^{ub} dz$$

where $ub = \sqrt{r^2 - x^2}$

Bibliography

[1] James W. Eastwood. Orthovec: A REDUCE program for 3-D vector analysis in orthogonal curvilinear coordinates. *Comp. Phys. Commun.*, 47(1):139–147, October 1987.

[2] James W. Eastwood. ORTHOVEC: version 2 of the REDUCE program for 3-D vector analysis in orthogonal curvilinear coordinates. *Comp. Phys. Commun.*, 64(1):121–122, April 1991.

[3] M . Speigel. *Vector Analysis*. Scheum Publishing Co., 1959.