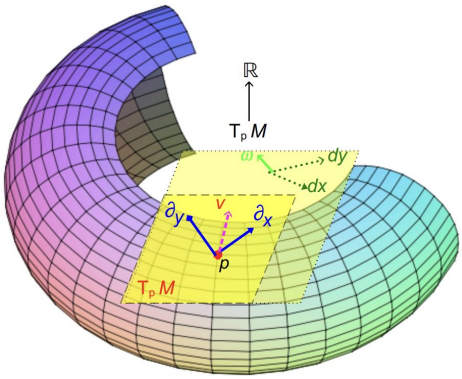
Exploring Math  with REDUCE

Differential Forms with REDUCE

A CARTAN Crash Course using `excalc`

with applications to Vector Analysis



Dr. Wolfgang G. Lindner
`dr.w.g.Lindner@gmail.com`
Leichlingen, Germany
2027

Contents

Preface
0.1 Why REDUCE? Why package `excalc`? . . . 3
0.2 Some Hints for the Use of REDUCE . . . 6

1 `TpM` : a mental image **7**

2 Tangent vectors are velocity vectors **8**
2.1 Exercises . . . 15

3 k - Forms : an overview **18**

4 0 - Forms **21**
4.1 Exercises . . . 23

5 1 - Forms and the Exterior Derivative **24**
5.1 Exercises . . . 27

6 Exterior Product and `eps` Tensor **28**
6.1 The `wedge` product aka the Exterior Product . . . 28
6.2 The LEVI-CEVITA `eps` tensor . . . 31

7 2 - Forms **36**
7.1 Exercises . . . 39

8 3 - Forms **40**
8.1 Exercises . . . 43

9 HODGE $\star$ operator **44**
9.1 Exercises . . . 49

10 `grad`, `curl`, `div`, `laplace` in Exterior Calculus **50**
10.1 Exercises . . . 58

11 Pullback **60**
11.1 Change of variables . . . 61
11.2 The pullback . . . 64
11.3 Definition of the Integral $\int_M \omega$ . . . 69
11.4 Exercises . . . 74

12 Integration on Manifolds aka Vector Analysis **76**
12.1 Integration with REDUCE . . . 76
12.2 Arc length . . . 78
12.3 Line integral . . . 83

12.4 Surface area 90
12.5 Surface integral 97
12.6 Volume integral 101

13 Generalised Stokes' Theorem **105**
13.1 Gradient Theorem 106
13.2 Green's Theorem 110
13.3 Stoke's Theorem 116
13.4 Gauss's Theorem 123

14 Bibliography **129**

0.1 Why REDUCE? Why package `excalc`?

A more advanced description of the well-known differential operators `grad-div-curl` can be provided using differential forms, as Andrew WINKLER gives so aptly in response to the question *How are a gradient, curl, divergence, Laplacian, etc. related to each other?*[1] on Quora:

> They are all constructed from the 'exterior derivative', together with the 'dot product' and its associated 'Hodge duality'.
>
> The exterior derivative of a function is a one form. You can use the dot product to convert that into a vector. That's the gradient.
>
> The exterior derivative of a one form is a two form. You can use Hodge duality to convert it to a one form, and then the dot product to make it a vector. That's the curl.
>
> The exterior derivative of a two form is a three form. You can use Hodge duality to convert it into a 0 form, that is a function. That's the divergence.
>
> In three dimensions the exterior derivative of a three form is 0.
>
> If you integrate the norm squared of the gradient of a function, you can integrate by parts to get an integral of the function times a differential operator applies to the function. That's the Laplacian.

This roughly outlines the aim of this little book.

The books by COLLIER [9] and BACHMAN [4] give a concrete introduction to the theory and praxis of differential forms. These book focusses on comprehension and illuminates mathematical concepts by pictures and worked examples to gain a working knowledge for the learner. This makes these books a perfect candidate for an learning environment at an 'action and process level' in the sense of the APOS Theory, cf. the short summary [25] and [26]. I also recommend the texts by BRYAN [8], LEBL [18], O'NEILL [36], SCHULZ [43] and for the German speaking readers the book by JÄNICH [16].

Rounding off in the sense of the APOS theory this booklet presents activities on the PC using the CAS REDUCE and in particular the package `excalc` by Eberhard SCHRÜFER. To give an action-process-playground for the material presented e.g. in those books, I have done accompanying examples and exercises using the language of the CAS REDUCE, which is very well suited in the context of differential forms using REDUCE`excalc`. However, this little book should also be readable independently of the books of COLLIER, BACHMAN, or at an advanced level, the book by NEEDHAM [37].

The calculations are therefore presented in an intermediate precise CAS-language, that lives between the slang, in which mathematical concepts are vivid and roughly explained and the high precision of the formal mathematical language, which is not so easy to grasp at a first attempt. The REDUCE language allows to make the mathematical concepts of

[1] A. Winkler, cf. relations

differential forms executable, to coin math processes (e.g. the concept of a differential forms) into 'run'able procedures and therefore to allow own experiments. The results of one's thinking in the CAS language is immediately returned to the screen and helps to check the right understanding. Also, intermediate calculations may be outsourced as programs to the CAS REDUCE.

To facilitate the transition from a classical vector-analytic treatment of the concepts under discussion to a representation based on differential forms, solutions to example problems are first presented using classical vector analysis description with the REDUCE package `avector` and afterwards repeated with the use of differential forms and the package `excalc`.

For the sake of comparability of a mathematical vs a computational solution, I have drawn most of the examples and exercises from calculus courses available online – specifically those from MARSDEN & WEINSTEIN [31] and e.g. OPENSTAX [35]. There, the reader can read a purely mathematically formulated solution in comparison.

It is recommended to look up a CAS REDUCE solution only after one's own solution steps. At a minimum, the learner should, at least in advance, create a plan of the solution in his mind. Then you can look at the REDUCE calculations and, if necessary, expand them into a detailed traditional mathematical computation.

Some remarks on the use of REDUCE`excalc`

The presented code in this booklet is 'tutorial' code, aimed at the novice learner who is willing to learn the theory and praxis of differential forms using the CAS REDUCE simultaneously. I have tried to program all concepts *as simple as possible*.

- To get a REDUCE script of this text click on a link like

   *Get Code.*

to view the corresponding script and (after copying parts of it) execute it in REDUCE. This is more safe w.r.t. *mark-copy-paste* of the `blue` code lines from this PDF file.

REDUCE is available for free to install on Windows, macOS and Linux. Please go to SourceForge , select the most recent snapshot directory, select the appropriate platform, then download and open the appropriate installer file. For further information, please see How to Obtain and Run REDUCE.

The code lines were tested on an `iMac 24"`, Apple M3, Tahoe 26.6.2 using Safari with REDUCE (CSL, rev 6864), 24-Aug-2024.

Acknowledgement. While using REDUCE and the differential form package `excalc` to write these notes I benefited greatly from the experience and expertise of E. SCHRÜFER. He also corrected or optimized some procedures, so I am very grateful for his friendly help and support with tips and hints.

Dr.F.J.Wright provided me with support in using the Reduce package `avector`, for which he has developed an enhanced version. He was also a great help with general questions regarding Reduce. Francis corrected a number of mistakes and errors; beyond these fixes, he also offered suggestions and contributions that significantly enhanced the readability and clarity of the work. I appreciate his willingness to engage in an exchange of ideas w.r.t. Reduce and his selfless, friendly assistance throughout the compilation of this project. His expertise regarding Reduce was of inestimable value to me.

I also ask sometimes the free AI Meta, Grok, Gemini, ChatGPD or Claude for help with coding. My experiences with this assistence are excellently summarized by the words of Viktor Todt from the Maxima community:

> Please tell the AI to keep its hands off the tensor packages unless it actually knows what it is doing... which it obvious doesn't, and I say this as someone who is very fond of AI-assisted coding, but with proper supervision and knowledgeable oversight.

In most cases there are no direct usable error-free AI solutions w.r.t. Reduce code, one has to look for misconceptions by the AI etc. and it is necessary, that the user of AI must have a solid understanding of the subject matter to isolate the problems with the presented code lines. Nevertheless, in the hand of a knowledgeable user the *AI assisted thinking and learning* is a great asset. The reader of this booklet is invited to try this out during the exercises offered. This way AI could offer valuable service and a scaffolding part in search of a potential solution.

Any feedback from the reader of this book is very welcome.

Wolfgang G. Lindner
Leichlingen, Germany
January 2027

0.2 Some Hints for the Use of REDUCE

1. You may paste single commands or the whole `*.txt` file with code lines from our scripts into the REDUCE [*Input Lines*].

   Advice: submit each of the command lines (separated by blank lines) in our scripts one after another to the REDUCE [*Input Lines*] and then press [enter].
   Observe and study the result for a good learning process.

2. The '*on demo*;' switch (for scripts & presentations)

   When on, REDUCE pauses after each command input from a file (with the exception of comments) until a `Return` is typed on the terminal.

   When you load a command file into REDUCE, the system normally executes all lines in immediate succession. The '`demo`' switch allows you to pause execution step-by-step:

   - ▷ *Activation*: Enter 'on demo;' at the command line (or place it at the beginning of your '.red' file).
   - ▷ *Function*: Once enabled, when you load a file using the command '`in "filename.red";`', REDUCE will read and display one statement at a time, then wait for you to press `Enter` (Return) before executing the next command. This is ideal for live demonstrations or for following along with calculations step-by-step.
   - ▷ *Deactivation*: Use '`off demo;`' to return to normal mode.

3. *Starting interactive lessons*

   If you would like to go through one of the included files step by step - for example, to help your mind to internalize the impact of the instruction, you can proceed as follows:

   ▷ start REDUCE.

   ▷ Write *on Demo* as the first line (command).

   ▷ Then read the file in using e,.g. '`in "swr11-2.txt";`'

4. STOUTEMYER's Interactive Lessons

   If you would like to try out REDUCE and learn its syntax, the installation directory typically includes seven interactive lessons (REDUCE Interactive Lessons by David R. STOUTEMYER): Depending on your installation, you can launch them using the command 'in "less1.red";' (up to 'less7.red'), cf. Lessons

1 TpM : a mental image

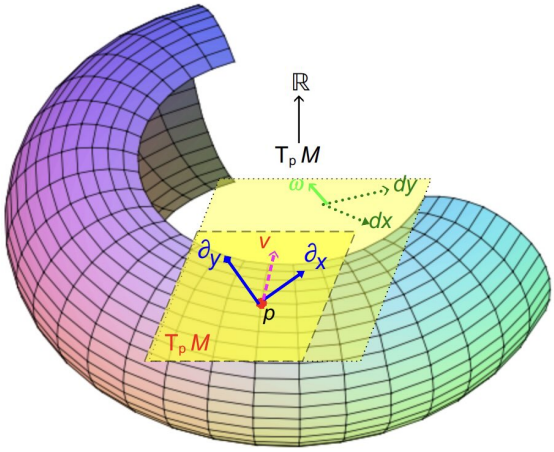


Figure 1:
M: manifold is part of a torus.
p: a position vector or point on M.
∂_x, ∂_y: the tangent(ial) plane vector basis.
dx, dy: the dual basis vectors of T_p^*M.
T_pM: here lives the tangent vectors alias the velocity vectors v.
T_p^*M: and at the same time the (differential) forms $\omega : T_pM \to \mathbb{R}$.

Here is the title page figure once again. Look at it for a while and enjoy it.
You should imagine: At each point p of a manifold M we may associate a tangent space T_pM, consisting of the velocity vectors v of curves on the manifold through point p. The 1-(differential)forms ω live also on T_pM, being mappings from T_pM to $\mathbb{R}$, noted T_p^*M.

To think in REDUCE[excalc] adapt the following notation:

LEXICON	*figure.1*	REDUCE excalc	
partial differentiation	$\partial_x \equiv \frac{\partial}{\partial x}$	`@ (f,t)`	in package `excalc`
dual basis of 1-forms	dx, dy	`d x, d y`	the blank in 'd x' is mandatory!

This sets up the scene for the following.

Now we calculate some velocity vectors to realize tangent vectors, i.e. elements of T_pM.

2 Tangent vectors are velocity vectors

We begin with two examples of tangent vectors to gain a visual mental image and a working knowledge of this concept. Imagine a tangent vector as a vector that is tangential at a given point to a curve, a surface or a higher dimensional space, i.e. a 'manifold'.

Example 1. (Tangent vector on a Parabola)
Imagine a mass point that travels along a parabolic path over time at a certain speed. If the mass point leaves the path at a certain position, it leaves the parabolic path *tangentially*. We denote the path by M ('manifold'), the position vector of the point at time t by $R(t)$ and the velocity by v.

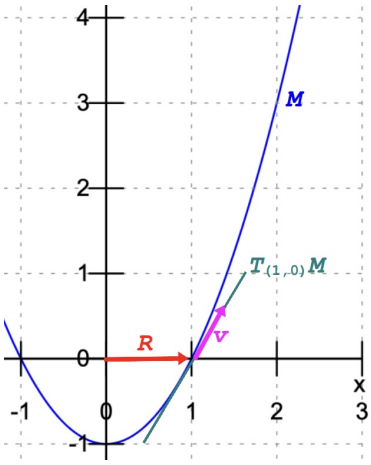


Figure 2: M: manifold is the parabola $\{(x, y) \in \mathbb{R}^2 | y = x^2 - 1\}$.
R: the position vector at $t = 1$ is $R(1) = (0, 1)$.
v: a general velocity vector is $R'(t) = (1, 2t)$.
$T_{(0,1)}M$: tangent line at point $P = R(1)$ on M.

Then the velocity vector is defined as follows:[2]

$$\mathbf{v} := \underbrace{\frac{\partial R}{\partial t}}_{\text{Math}} = \begin{matrix} \mathrm{df}(R,t) & : \text{in core REDUCE} \\ \underbrace{@(R,t)}_{\text{REDUCE}} & : \text{in package } \texttt{excalc} \end{matrix} \tag{2.1}$$

Task:

a. Calculate the velocity vector v in Fig.2. first mathematically.

b. Calculate the velocity vector v in Fig.2. using core REDUCE.

c. Calculate the velocity vector v in Fig.2. using package `excalc` of REDUCE.

[2] $\frac{\partial R}{\partial t}$ is often noted as $\partial_t R$ or $\dot{R}(t)$.

Solution:

ad a.: we solve the problem using elementary calculus.
We express the path R by his coordinate functions and derivate these.

1. $R(t) := (x(t), y(t))$ where $x(t) = t$ and $y(t) = t^2 - 1$ — coordinate functions
2. $\dot{R}(t) = (\dot{x}(t), \dot{y}(t)) = (1, 2t)$ — componentwise derivation
3. $\dot{R}(t = 1) = (1, 2) = v$ — read off the solution; v in the sketch is too short ;)

Result: Let $R : I \to \mathbb{R}^2$ with $t \mapsto (t, t^2)$ be differentiable on the open interval $I \subset \mathbb{R}$. Then $R'(t) = \frac{\partial}{\partial t}(t, t^2 - 1) = (1, 2t)$ and $R'(1) = (1, 2) = v$ is the *tangential vector* at the time $t = 1$ resp. at the point $P = (0, 1)$. □

ad b.: we solve the problem using core REDUCE following the steps from **a.**
`df` is the standard differentiation operator in *core* REDUCE, for a review see [50].
No external packages are required for differentiation.[3]

```
% SOLUTION IN core REDUCE

% 1. Define the component functions of R=(t, t^2-1)
R1 := t_$
R2 := t_^2-1$

% 2. Differentiate components with respect to t_
dR1 := df(R1,t_);
dR2 := df(R2,t_);

% 3. Calculate the tangent vector v substituting 1 for  t_
v := sub(t_=1,{dR1,dR2});

% FIRST ALTERNATIVE SOLUTION

% 1. Define the component functions as a list
R := {t_, t_^2-1};

% 2. Differentiate each component with respect to t_
%    and collect the results in a new list named v
v  := for each el in R collect df(el, t_);
v1 := sub(t_=1,v);   % evaluate v at 1
```

[3] In core REDUCE the symbol t is a reserved identifier for the boolean type `true`. We recommend to use the variable name $t_$ instead or use s.

```
% SECOND ALTERNATIVE SOLUTION

% here is a one-liner using operator map()
R := {s, s**2}$
v := sub(s=1, map(df(~w, s), R));
```

 Get *code*.

ad c.: In REDUCE[excalc], to calculate the derivative of the curve $R(t) = (t, t^2 - 1)$ is to declare the component functions x and y as 0-forms (=usual functions), assign them the values t and t^2-1 [4].respectively, and then apply the partial differentiation operator @ [5]to them:

```
% USING excalc PACKAGE OF REDUCE

% the following command _must_ be executed first in a input line for itself
load_package excalc;

% declare component functions x and y as so-called 0-forms by  ..=0
pform t=0, x=0, y=0;

% set dependencies
fdomain x=x(t), y=y(t);

% define the parametrization
x := t;
y := t**2;

% differentiate using excalc's partial differentiation operator @
@(x,t);
@(y,t);

% calculate tangent vector v
v := sub(t=1, {@(x,t),@(y,t)});
```

REDUCE output:

```
@(y,t);
```

$$2t$$

```
% calculate tangent vector v
v := sub(t=1, {@(x,t),@(y,t)});
```

$$v := \{1\ ,\ 2\}$$

 Get *code*.

[4]Remember: in `excalc` the symbol $\wedge$ is reserved for the so-called 'wedge' product, therefore the exponentiation is written in old style with two stars in a row '`**`', i.e. we note t^2 as `t**2`

[5]corresponds to the standard REDUCE operator *df*

Example 2. (Tangent vector on a Sphere) The unit sphere S^2 is a 2-dimensional manifold ('surface') in its surrounding space $\mathbb{R}^3$.
Let's describe the sphere $S^2 := \{(x, y, z) \in \mathbb{R}^3 | x^2+y^2+z^2 = 1\}$ by means of a parametrization using the position vector $R(\theta, \phi)$.

Let $R : U \to \mathbb{R}^3$ be defined by

$$(\theta, \phi) \mapsto (\cos(\theta)\cos(\phi), \cos(\theta)\sin(\phi), \sin(\theta)) \tag{2.2}$$

and differentiable on an open subset $U \subset \mathbb{R}^2$ with $(\theta, \phi) \in U$.
Then we have two tangent vectors: $\mathbf{Z_1} := \frac{\partial R}{\partial \theta}$ and $\mathbf{Z_2} := \frac{\partial R}{\partial \phi}$, which are the *tangent vectors* at the point $R(\theta, \phi) \in S^2$ with coordinates (θ, ϕ) and span the tangent plane $T_P M$.

Task:
a. Calculate the tangent vectors at an general point $P = R(\theta, \phi)$.
b. Calculate the tangent vectors at the special point $P = R(45°, 0)$.

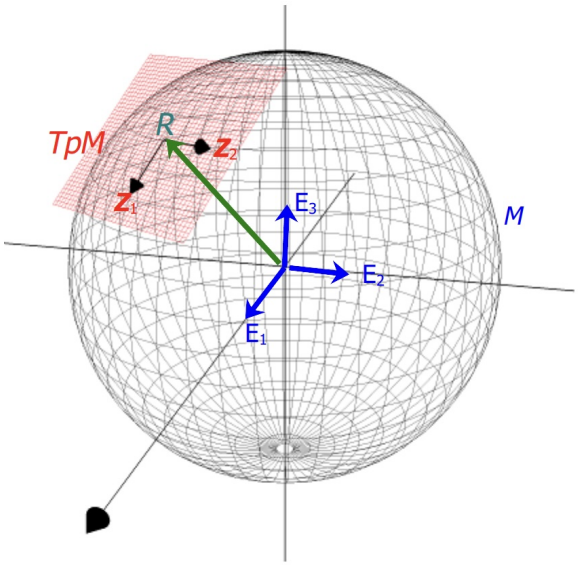


Figure 3: (E_1, E_2, E_3): canonical Euclidian basis in $\mathbb{R}^3$.
(Z_1, Z_2): basis of tangent plane $T_P S^2$ at $R(\theta, \phi) \in S^2$.
R: the position vector from the origin O to $R(\theta, \phi)$.
M: the manifold $S^2 \subset \mathbb{R}^3$, i.e. the sphere of radius 1 with origin O.

Solution:

1st mathematically.
a:

$$\begin{aligned}\mathbf{Z_1} &= \frac{\partial R}{\partial \theta} = \frac{\partial}{\partial \theta}(\cos(\theta)\cos(\phi), \cos(\theta)\sin(\phi), \sin(\theta)) \\ &= (-\cos(\phi)\sin(\theta), -\sin(\phi)\sin(\theta), \cos(\theta))\end{aligned}$$

$$\begin{aligned}\mathbf{Z_2} &= \frac{\partial R}{\partial \phi} = \frac{\partial}{\partial \phi}(\cos(\theta)\cos(\phi), \cos(\theta)\sin(\phi), \sin(\theta)) \\ &= (-\sin(\phi)\cos(\theta), \cos(\phi)\cos(\theta), 0))\end{aligned}$$

b:

$\mathbf{Z_1} \mid (\frac{\pi}{4}, 0) = (\frac{-1}{\sqrt{2}}, 0, \frac{1}{\sqrt{2}})$
$\mathbf{Z_2} \mid (\frac{\pi}{4}, 0) = (0, \frac{1}{\sqrt{2}}, 0)$

2nd solution with REDUCE.

```
% .......... WITH core REDUCE

R := {cos(theta)*cos(phi), cos(theta)*sin(phi),  sin(theta)}$
Z1 := {df( first R, theta), df( second R, theta), df( third R, theta)};
Z2 := {df( first R, phi), df( second R, phi), df( third R, phi)};
sub( {theta=pi/4, phi =0}, Z1);
sub( {theta=pi/4, phi =0}, Z2);

% .......... WITH jacobian of PACKAGE linalg

load_package linalg;

% 1. Define your list of components (i.e. the vector function)
R := {cos(theta)*cos(phi), cos(theta)*sin(phi),  sin(theta)}$

% 2. Define the list of variables
vars := {theta, phi};

% relax the notation
define jacobian = mat_jacobian;

% 3. call procedure jacobian
% the solutions Z1 and Z2 are the columns of the jacobian
jacobian(R, vars);
```

 Get *code*.

REDUCE output:

```
load_package linalg;

R := {cos(theta)*cos(phi), cos(theta)*sin(phi),  sin(theta)}$

vars := {theta, phi};
```

$$\text{vars} := \{\theta, \phi\}$$

```
define jacobian = mat_jacobian;

jacobian(R, vars);
```

$$\begin{pmatrix} -\cos(\phi)\sin(\theta) & -\cos(\theta)\sin(\phi) \\ -\sin(\phi)\sin(\theta) & \cos(\phi)\cos(\theta) \\ \cos(\theta) & 0 \end{pmatrix}$$

$\mathbf{3}^{rd}$ solution using package `excalc`:

```
load_package excalc;

% declare the coordinates x, y and z as 0-forms
pform x=0, y=0, z=0;

% set dependencies in the function domain
fdomain x=x(theta,phi), y=y(theta,phi), z=z(theta,phi);

% define the parametrization via component functions
x := cos(theta)*cos(phi);
y := cos(theta)*sin(phi);
z := sin(theta);

% differentiate using excalc's partial differentiation operator @
Z1 := {@(x,theta), @(y,theta), @(z,theta)};
Z2 := {@(x,phi), @(y,phi), @(z,phi)};

% calculate tangent vectors v
sub( {theta=pi/4, phi =0}, Z1);
sub( {theta=pi/4, phi =0}, Z2);
```

Get *code*.

Remark. A tangent vector is a so-called *contravariant vector* and should conventionally better written in column form, e.g.

$$\mathbf{Z_1} = (-\cos(\phi)\sin(\theta), -\sin(\phi)\sin(\theta), \cos(\theta))^t = \begin{bmatrix} -\cos(\phi)\sin(\theta) \\ -\sin(\phi)\sin(\theta) \\ \cos(\theta) \end{bmatrix}$$

Admittedly, we've been a bit lax in this regard up until now.

To sum up the notation:

LEXICON	*Math*	REDUCE	
partial differentiation of $f(x,y)$	$f'(t) = \frac{\partial f}{\partial t}$	`df(f,t)`	in core REDUCE
		`@ (f,t)`	in package `excalc`

Remark. (Notation of tangent vectors in REDUCE$^{\texttt{excalc}}$)
1. Vectors in $\mathbb{R}^3$ are usually identified with its coordinates and witten e.g. $v = (1,2,3)$. This is a bit abusive and sometimes irritating, because this notation is only correct in the canonical basis (E_1, E_2, E_3) of $\mathbb{R}^3$, where e.g. $E_1 = (1,0,0)$, see fig.3.
Therefore we should write precisely

$$v = (1,2,3) \equiv 1 \cdot E_1 + 2 \cdot E_2 + 3 \cdot E_3 \overset{\text{often}}{=} 1 \cdot \mathbf{i} + 2 \cdot \mathbf{j} + 3 \cdot \mathbf{k} \qquad \in \mathbb{R}^3 \qquad (2.3)$$

2. Analog we declare tangent vectors $v \in T_P M$ in REDUCE$^{\texttt{excalc}}$ using the declaration `tvector` often with respect to a choosen coordinate system.

```
load_package excalc;

% basis of TpM analog to basis E1, E2, E3 in R^3
coframe e(1) = d x, e(2) = d y, e(3) = d z;

% declare the coordinate functions x, y, z as 0-forms
pform x=0, y=0, z=0;

% expressions containing differentials of the coordinate functions
% will be transformed into expressions of the basis one-forms
tvector v;
v:=1*@ x + 2*@ y + 3*@ z ;
```

REDUCE output:

```
v:=1*@ x + 2*@ y + 3*@ z ;
```

$$v := \partial_x + 2\partial_y + 3\partial_z$$

Get *code*.

2.1 Exercises

Exercise 1. (tangent vector on a circle)
The unit circle is a 1-dimensional manifold ('curve') in $\mathbb{R}^2$.
We recommend to describe this circle $S^1 := \{(x,y) \in \mathbb{R}^2 | x^2+y^2 = 1\}$ via a parametrization using the position vector $R(t) := (\cos(t), \sin(t))$.
- Calculate the tangent vector at $P = R(45°)$ with/out REDUCE.

Get *code*.

You should check your solution at fig.4.

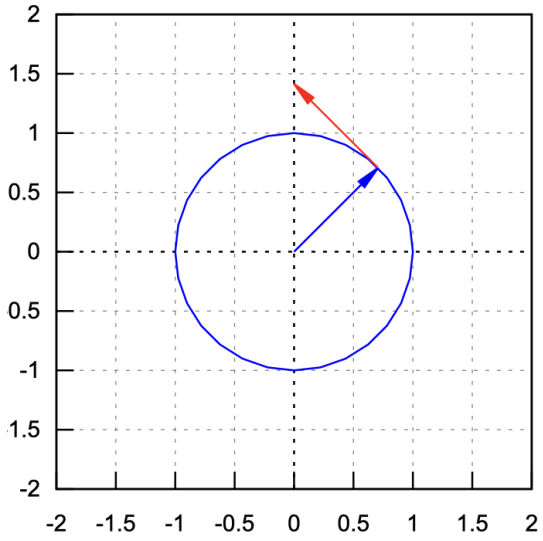


Figure 4: $R(\pi/4)$: position vector at time $t = \pi/4$ pointing on S^1.
$\partial_t R$: tangent vector $R'(\pi/4)$ at $R(\pi/4) \in S^1$.

Exercise 2. (user-defined function *jacobian* for REDUCE)
Write a user-defined function `jacobian` and
a. calculate again the tangent vectors Z_1 and Z_2 from example 2.
b. calculate the tangent vector from exercise 1.

Get *code*.

Exercise 3. (the directional derivative for REDUCE, cf. [11, p.4])
Let U denote an open subset in $\mathbb{R}^n$ and let $f : U \to \mathbb{R}^m$ with $f = (f_1, ..., f_m)$ components. Then we define for $X \in U$ and $v \in \mathbb{R}^n$ the *directional derivative of f in the direction v at X* to be

$$D_v f(X) := \lim_{t \to 0} \frac{f(X + t \cdot v) - f(X)}{t} \tag{2.4}$$

a. Approximate $D_v f(X)$ using formula (2.4) for the data set
   - $F : \mathbb{R}^4 \to \mathbb{R}^4$ with $F(x,y,z,w) := (x^2 y, xyz, x^2 + y^2 + zw^2)$,

- point $X := (1, 2, -1, -2)$ and vector $v = (0, 1, 2, -2)$ using $t = 0.0001$.

What should be the exact value of $D_v f(X)$? (cf. [11, p.5])

b. Write a user-defined function `DvF`.

c. Calculate `DvF(X)` using b. with the data set from a.

d. Calculate `DvF(X)` with the data from a. using REDUCE[excalc].

Solution.

In this exercise, we anticipate some concepts that we will discuss in the next sections. Therefore we present in advance the following *Solution* using package `excalc`, which was contributed by E. SCHRÜFER:

```
load_package excalc;

pform {x, y, z, w} = 0;

% F as a 1-form expressed in the basis d_x, d_y and d_z
f := x**2*y*d x + x*y*z*d y + (x**2 + y**2 +z*w**2)*d z;

% v as tangential vector expressed in basis @_x, @_y, @_z, @_w
v := vx*@ x + vy*@ y +vz*@ z + vw*@ w;

% then the directional derivative in direction v is
dvf := v _| (d (@ x  _|  f))*d x +
       v _| (d (@ y  _|  f))*d y +
       v _| (d (@ z  _|  f))*d z ;

% specialized for vector v: vx=0, vy=1, vz=2, vw=-2
dvf0 := sub(vx=0, vy=1, vz=2, vw= -2, dvf);

% and for point x=1, y=2, z= -1, w= -2
sub(x=1,y=2,z=-1,w=-2, @ x _| dvf0);
sub(x=1,y=2,z=-1,w=-2, @ y _| dvf0);
sub(x=1,y=2,z=-1,w=-2, @ z _| dvf0);
```

Comment. Regarding the directional derivative of a function, it is given in the exterior calculus by the interior product of the direction vector v and the differential df of the function, i.e., $v \lrcorner df$. Thereby one interprets F as the definition of a 1-form.

 Get *code*.

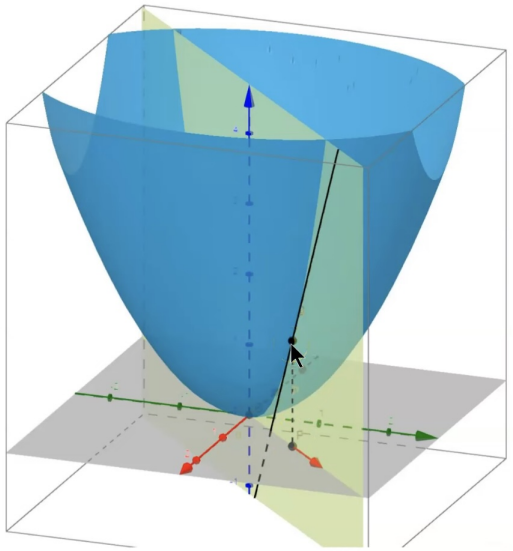


Figure 5: A visualisation of the directional derivative, see the video at *DvF(X)*

To quote *mathinsight*: A directional derivative is the slope or rate of change of a surface $z = f(x, y)$ when moving in a specific, arbitrary direction rather than just parallel to the x or y axes.
Imagine a 3D surface (like a bowl or a hill). If you pick a point on the surface and point along a unit direction vector $\vec{u}$, a vertical plane cuts straight down through the surface along that direction vector. That vertical slice creates a 2D curve (a parabola-like shape) on the cutting plane. The directional derivative is simply the slope (the derivative) of that curve right at your point of interest.

Exercise 4. (from [7, p.177])
Calculate the directional derivative of the scalar function

a. $f(x, y, z) = x^2 + y^2 + z^2$ at the point $(1, 2, 3)$ in direction $\vec{a} = (1, 3, 5)$. 44

b. $f(x, y, z) = e^{xz} \cos y$ at the point $(1, \pi, 2)$ in direction $\vec{b} = (1, 1, 2)$. $-4e^2$

Get *code*.

Exercise 5. (from [4, p.12])
Suppose $f(x, y) = xy^2$. Compute the slope of the tangent to $f(x, y)$ at the point $(2, 3)$ in the direction $(1, 5)$. – BACHMAN gives a nice visualization on p. 11. 69
Get *code*.

3 k - Forms : an overview

$$\omega : (TpM)^k \longrightarrow \mathbb{R} \tag{3.1}$$

multilinear and alternating

k=0: $\omega : M \subset \mathbb{R} \longrightarrow \mathbb{R}$ is simply a differentiable scalar function.
It eats a number v and returns a number $\omega(v)$. Example: $\omega(v) = v^2 + 2v - 1$.

k=1: $\omega : TpM \longrightarrow \mathbb{R}$ is a 1-form.
It eats a (tangent) vector and returns a number $\omega(v)$.
A typical 1-form looks like $\omega = 3x\,dx - 2\,dy + 4\,dz$.

k=2: $\omega : TpM \times TpM \longrightarrow \mathbb{R}$ is a 2-form.
It eats two (tangent) vectors and returns a number $\omega(v, w)$.
A typical 2-form looks like $\omega = 2x\,dx \wedge 3\,dz$.[6]

k=3: $\omega : (TpM)^3 \longrightarrow \mathbb{R}$ is a 3-form.
It eats 3 vectors and returns a number $\omega(u, v, w)$.
A typical 3-form looks like $\omega = -3y\,dx \wedge 2\,dz \wedge -1\,dy$.

k=...

Definition. The set of k-forms on a manifold M are denoted as $\Omega^k M$,
i.e. read $\omega \in \Omega^k \mathbb{R}^2$ as 'ω is a k form on $\mathbb{R}^2$ '.

Differential forms and tangent vectors live on manifolds; they interact at a specific point to yield a numerical result. We denote a 1-form acting on a vector v as $\omega(v)$[7] using lower case Greek letters, and a 2-form acting on two vectors u and v as $\omega(u, v)$. Differential forms share the properties of *multi-linearity* and *alternating* behavior with the familiar determinant; let us therefore begin by examining the determinant function *det*.

Example 3. (the determinant as multilinear alternating function e())
We check the characteristics of *multi-linearity* and *alternating* by core REDUCE.

```
operator e;                  % operator e() with properties of 2x2 det()
noncom e;                    % e is declared noncommutative     *  e1  e2
for all j let e(j)**2 = 0;                                  %  e1   0   1
let e(2)*e(1) = -e(1)*e(2);                                 %  e2  -1   0

v := (a*e(1)+b*e(2));
```

[6] This 2-form is constructed using the so-called wedge-product $\wedge$, see next section. It is sometimes also called the GRASSMANN product.

[7] we will state this 'action' more precisely and explict in §4 and §5.

```
w := (c*e(1)+d*e(2));

v * w;
w * v;            % switch sign, i.e. 'alternating'

v := 3*e(1)+1*e(2);
w := 1*e(1)+4*e(2);

v*w;              % changing order change the sign of the result
w*v;

det mat((3, 1),   % compare with build-in determinant
        (1, 4));
```

REDUCE output:

```
v := 3*e(1)+1*e(2);
```

$$v := e(2) + 3e(1)$$

```
w := 1*e(1)+4*e(2);
```

$$w := 4e(2) + e(1)$$

```
v*w;
```

$$11\,e(1)\,e(2)$$

```
w*v;
```

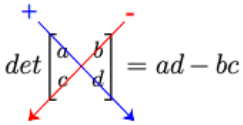
$$-11\,e(1)\,e(2)$$

```
det mat((3, 1),
        (1, 4));
```

$$11$$

Get *code*.

Comment. By the way, we have the following *mnemonic pattern*, the so-called *rule of* LEIBNIZ, to calculate the determinant of a 2×2 matrix, cf. [24, p.17].



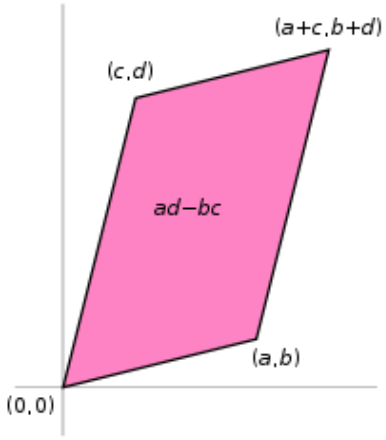


Figure 6: The LEIBNIZ formula for the determinant $det\begin{bmatrix} a & b \\ c & d \end{bmatrix} = ad - bc$ of a 2×2 matrix $M = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ allows a geometrical interpretation as *oriented* area of a parallelogram with vertices $(a, b), (c, d)$.

Exercise 6. w.r.t. Fig.6

a. Verify: Let the columns $A = \overrightarrow{OA} = (a, b)$ and $C = \overrightarrow{OC} = (c, d)$ of matrix M be drawn as the sides of a parallelogram[8] $OABC$ with vertices at $O = (0, 0), A = (a, b), B = (a + c, b + d)$, and $C = (c, d)$. Verify by an elementary geometric argument, that *the (oriented) area of parallelogram* $OABC$ *equals* $det(A)$, i.e.

$$area(OABC) = |det(\overrightarrow{OA}, \overrightarrow{OC})|$$

b. Verify the bilinearity of the operator $e()$ in the REDUCE example.
Get *code*.

[8] We cite the figure from `https://en.wikipedia.org/wiki/Determinant`

4 0 - Forms

Q: What is a typical 0-form?
A: a differentiable scalar function $f : \mathbb{R}^n \to \mathbb{R}$.

0-forms can be differentiated, integrated and plotted. They are theme of elementary calculus. In the context of REDUCE`excalc` 0-forms are declared using the key word `pform`.[9]

Remark. The set of 0-forms on a manifold M are noted as $\Omega^0 M$,
i.e. read $\omega \in \Omega^0\mathbb{R}^2$ as 'ω is a zero form on $\mathbb{R}^2$'.

Definition (*action of a 0-form*)**.**
The '*action* of a 0-form f on vector v' is nothing but the *evaluation* of v by f, i.e.

$$\text{action of } f \text{ on } v := f(v) \tag{4.1}$$

Example 4. (partial differential and integral of a 0-form in REDUCE`excalc`)
Let $f : \mathbb{R} \to \mathbb{R}, x \mapsto x^2 - 1$ be a 0-form, i.e. $f \in \Omega^0\mathbb{R}$.

a. Calculate the derivation of f and it's value at $x = 2$.

b. Calculate the integral of f in general form $\int f$ (indefinite integral) and the definite integral $\int_a^b f$ on the manifold (interval) $M := [2, 3]$ using function `int` and do a numeric integration using `num_int`.

c. Plot f over M.

Use package `excalc` for the solution.

Solution.

1st we solve this simple problem using elementary calculus to compare with REDUCE`excalc`.

1. $f(x) := x^2 - 1$ — scalar function f alias 0-form
2. $f'(x) = 2x$ and $f'(2) = 4$ — derivation
3. $\int f = \frac{1}{3} \cdot x^3 - x$ — indefinite integral of f
4. $\int_2^3 f = [\frac{1}{3} \cdot x^3 - x]_2^3 := \frac{1}{3} \cdot 3^3 - 3 - (\frac{1}{3} \cdot 2^3 - 2) = \frac{16}{3}$ — definite integral of f

[9] The name 'pform' comes from the name PFAFF'ian form in the mathematical literature.

2nd we solve this problem using REDUCE$^{\texttt{excalc}}$ to learn the notation.

```
% FIRST do the following command on a line for itself (!)
load_package excalc;

% THEN declare the scalar function f and its dependencies
pform f = 0;
fdomain f = f(x);
f := x**2-1;          % do NOT write x^2

% do the derivative using the partial operator @
fp := @(f,x);         % f prime i.e. f'(x)
sub(x=2, fp);         % substitute 2 for x in f'(x)

% integrate using core REDUCE
int (x**2-1, x);
int (x**2-1, x, 2, 3);

% numerical integration, see manual 17.4
num_int(f, x = (2 .. 3));

plot(x**2-1, 0, x=(0 .. 5), y=(-2 .. 10),  title="0-form f");
```

REDUCE$^{\texttt{excalc}}$ output:

```
fp := @(f,x);
```

$$\text{fp}:= 2x$$

```
sub(x=2, fp);
```

$$4$$

```
int (x**2-1, x);
```

$$\frac{x\left(x^2-3\right)}{3}$$

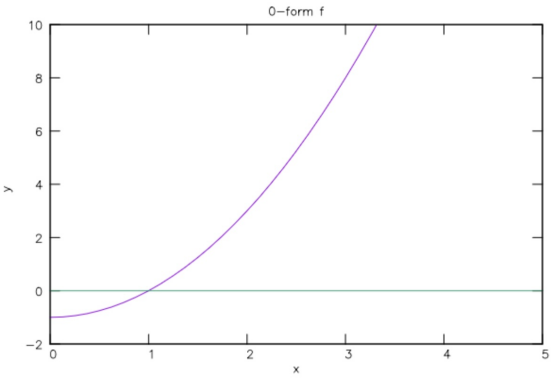
```
int (x**2-1, x, 2, 3);
```

$$\frac{16}{3}$$

```
load_package numeric;

num_int(f, x = (2 .. 3));
```

$$5.33333333333$$



Get *code*.

4.1 Exercises

Exercise 7. (0-form on $\mathbb{R}^2$)
Let $f : \mathbb{R}^2 \to \mathbb{R}, (x, y) \mapsto x^2 \cdot y$ be a 0-form.

a. Calculate the partial derivatives of f and it's value at $(x, y) = (2, 3)$ using `excalc`.

b. Calculate the integral of f in general form $\int f$ (indefinite integral) and the definite integral $\int_a^b f$ on the manifold (rectangle) $M := \{(x, y) \in [0, 4] \times [1, 5] \mid x + y = 2\}$ using function `int` and do a numeric integration using `num_int`.

c. Plot f over M.

Get *code*.

Exercise 8. (0-form on $\mathbb{R}^3$)
Let $f : \mathbb{R}^3 \to \mathbb{R}, (x, y, z) \mapsto x \cdot y \cdot z$ be a 0-form.

a. Calculate the partial derivatives of f and it's value at $(x, y, z) = (1, 2, 3)$ using `excalc`.

b. Calculate the integral of f in general form $\int f$ (indefinite integral) and the definite integral $\int_a^b f$ on the manifold (parallelepiped) $M := \{(x, y, z) \in [0, 4] \times [1, 5] \times [2, 1] \mid x + y + z = 2\}$ using function `int` and do a numeric integration using `num_int`.

Plot f over M not possible (would be a 3-dim submanifold of $\mathbb{R}^4$.

Get *code*.

Exercise 9. (0-form on $\mathbb{R}^2$ in ploar coordinates)
Let $h : \mathbb{R}^2 \to \mathbb{R}, (r, \varphi) \mapsto r \cdot \varphi$ be a 0-form.

a. Calculate the partial derivatives of h and it's value at $(r, \varphi) = (2, pi)$ using `excalc`.

b. Calculate the integral of h in general form $\int f$ (indefinite integral) and the definite integral $\int_a^b f$ on the manifold (rectangle) $M := \{(r, \varphi) \in [0, 2 \times [0, pi] \mid r + \varphi = 2\}$ using function `int` and do a numeric integration using `num_int`.

c. Plot h over M.

Get *code*.

5 1 - Forms and the Exterior Derivative

> The exterior derivative of a function is a one form. You can use the dot product to convert that into a vector. That's the gradient. [10]

Q: What is a typical 1-form?
A: the gradient `grad(f)` of a 0-form $f : \mathbb{R}^n \to \mathbb{R}$
or the work of a force on the displacement, cf. [3, p.164].

To realize A. we need the concept of the exterior derivative d. Fortunately, we do not need to memorize an explicit difficult formula for the exterior derivative, because the exterior calculus package `excalc` handles this definitional work for us.

Fact 1. (The exterior derivative d as degree increasing linear map)

$$\begin{aligned} d : \Omega^k M &\longrightarrow \Omega^{k+1} M \\ \omega &\mapsto d\omega \ =? \end{aligned} \tag{5.1}$$

with (1) $d(a * \omega + b * \nu) = a * d\omega + b * d\nu$ for all $a, b \in \mathbb{R}$ and $\omega, \nu \in \Omega^k M$ (Linearity)
and (2) $d(d\alpha) = 0$ for all $\alpha \in \Omega^k M$. (Nilpotency) [11]

Fact 2. (fundamental relations)
(3) The canonical basis of the space $\Omega^1\mathbb{R}^n$ are the 1-forms $dx^1, dx^2, ..., dx^n$.[12]
(4) The basis 1-forms dx, dy in $\mathbb{R}^2$ pick out the v_1, v_2 components of a tangent vector v.
(5) We have the *duality relations* of natural base elements:

$$dx^i(\partial_{x^j}) = \delta^j_i \tag{5.2}$$

where ∂_{x^j} are the coordinate (tangent) basis vectors in $T_p\mathbb{R}^n$.
Expression (5.2) means the 'action' of the 1-form dx^i on the tangent vector ∂_{x^j}, see below.

Remark. 1. The set of 1-forms on a manifold M are noted as $\Omega^1 M$,
i.e. read $\omega \in \Omega^1\mathbb{R}^2$ as 'ω is a one form on $\mathbb{R}^2$'.
2. The exterior derivative d is a fundamental operator that extends the concept of the differential of a function (aka 0-form) to higher-degree differential forms. d acts as a mapping from the space of k-forms to $(k+1)$-forms, systematically unifying the core operators of vector calculus i.e. gradient, curl, and divergence into a single coordinate-independent framework.
3. 1-forms are declared in `excalc` using the key word `pform` omega `=1`;.
4. The exterior derivative of a 0-form f (i.e. a smooth function) is its total differential df.

---

[10] A. Winkler, cf. relations
[11] where the ? stands for a complicated formula.
[12] Remember: the superscripts $...^1, ...^2, ...$ are indices, not exponents.

Definition (*action of a 1-form*).
The action $\omega(v)$ of a 1-form ω on a tangent vector v is defined by the *evaluation* of v by ω, i.e.

$$\text{action of } \omega \text{ on } v := \omega(v) := v \lrcorner \omega \tag{5.3}$$

∘ The mathematical *evaluation notation* $dx^i(\partial_{x^j}) = \delta^j_i$ in (5.2), here $dx(\partial_x) = 1, dx(\partial_y) = 0$ etc. is represented in `excalc` by the diphthong `_|` (underscore or-bar), i.e. using the *inner product* operator (aka *insertion* operator) '$\lrcorner$ '. If the exterior form is an exterior product, the inner product is carried through any factor.

Example 5. (action of ω on v in REDUCE$^{\texttt{excalc}}$)
Suppose $\omega = 2dx + 3dy$, then $\omega(-1, 2) = 2 \cdot (-1) + 3 \cdot 2 = 4$.

In REDUCE$^{\texttt{excalc}}$:

```
load_package excalc;
pform {x, y} = 0;
pform w=1;
w := 2 * d x + 3 * d y;
tvector v;
v := -1*@ x + 2*@ y
v  _|  w;
```

```
v := -1*@(x) + 2*@(y);
```

$$v := -\partial_x + 2\partial_y$$

```
v _| w;
```

$$4$$

Get *code*.

Example 6. (the basic duality relations (5.2) in REDUCE$^{\texttt{excalc}}$)
Verify the duality relations (5.2), fact (5) for the Cartesian coordinate basis in $\mathbb{R}^2$.

Solution.

```
load_package excalc;
pform {x, y} = 0;
@ x  _|  d x;
@ x  _|  d y;
@ y  _|  d y;
```

REDUCE$^{\texttt{excalc}}$ output:

```
@ x _| d y;
```

$$0$$

```
@ y _| d y;
```

$$1$$

Get *code*.

Example 7. (the gradient of a 0-form in REDUCE$^{\texttt{excalc}}$)
Let $f : \mathbb{R} \to \mathbb{R}, x \mapsto x^2 + 2x - 1$ be a 0-form, i.e. $f \in \Omega^0\mathbb{R}$.
a. Calculate grad(f) with `excalc`.
b. Calculate grad$_P(f)$ at unit $u = 1$ in position $p = 2$ with `excalc`.

Solution.

```
load_package excalc;

coframe e(1) = d x;    % 1. calculate w.r.t. basis e1 one-form (the coframe)
pform f = 0;           % 2. declare f as a 0-form (scalar function)
fdomain f = f(x);      % 3. tell EXCALC that f depends on x
f:=x**2+2*x-1;         % 4. define our explicit function

d f;                   % 5. calculate the gradient of f in R^1
1*@ x _| d f;          %    along unit tangent vector v=(1) of R^1
sub(x=2, 1*@ x _| d f);    % at point p=2
```

REDUCE output:

```
d f;
                                   2 e^1 (x + 1)
1*@ x _| d f;
                                     2 (x + 1)
sub(x=2, 1*@ x _| d f);
                                          6
```

Get *code*.

Comment. Operator d applied to the 0-form f gives exactly the 1-form $df = \frac{\partial f}{\partial x} dx$. The coefficient of df (with respect to the coframe $e1$) is exactly the gradient component. The coefficient $2x + 2$ is the single component of grad f, since dx forms the single element of the standard coframe of Euclidean space $\mathbb{R}^1$.

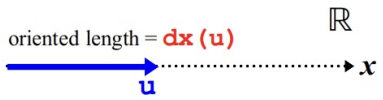


Figure 7: The evaluation formula $dx(u) \equiv u \lrcorner\, dx$ interpreted as the determinant $det\big[u\big] = u$ of 1×1 - matrix $\big[u\big]$, i.e. as the *oriented* length of u on the number line $\mathbb{R}^1$.

5.1 Exercises

Exercise 10. (basis 1-forms and basis tangent vectors)
Let $v = (3, -1, 6) \equiv 3 \cdot \partial_x - \partial_y + 6 \cdot \partial_z$ be a tangent vector of $T_p\mathbb{R}^3$.
Let $\omega = 5dx + 2dy + 4dz$ be a 1-form of $\Omega_p\mathbb{R}^3$.
a. Calculate the action of ω on v, i.e. $\omega(v)$ only using duality and linearity of ω.
b. Calculate the acting of ω on v using `excalc`.
ad a.: we solve a.

$$\begin{aligned}\omega(v) &\overset{p.d.}{=} (5dx + 2dy + 4dz) \lrcorner (3 \cdot \partial_x - \partial_y + 6 \cdot \partial_z) \\ &= 5dx(3 \cdot \partial_x) - 5dx(-\partial_y) + 5dx(6 \cdot \partial_z) + 2dy(3 \cdot \partial_x) + \dots \\ &= 15 \cdot dx(\partial_x) + 0 + 0 + 0 + 2 \cdot dy(\partial_y) + 4 \cdot dz(\partial_z) = 37\end{aligned}$$

ad b.: we solve this problem using REDUCE`excalc` to learn the notation:
Get *code*.

Exercise 11. (the gradient of a 0-form in $\mathbb{R}^2$)
Let $f : \mathbb{R}^2 \to \mathbb{R}, (x, y) \mapsto x^2 - y^3$ be a 0-form, i.e. $f \in \Omega^0\mathbb{R}^2$.
a. Calculate $\text{grad}(f)$ with `excalc`.
b. Calculate grad_v at tangent vector $v = (1, 1)$ and point $P = (2, 3)$ with `excalc`.
Get *code*.

Exercise 12. (the gradient of a 0-form in $\mathbb{R}^3$)
Let $f : \mathbb{R}^3 \to \mathbb{R}, (x, y, z) \mapsto x \cdot y^2 \cdot z^3$ be a 0-form, i.e. $f \in \Omega^0\mathbb{R}^3$.
a. Calculate $\text{grad}(f)$ with `excalc`.
b. Calculate $\text{grad}_v(f)$ at $v = (-1, 0, 1)$ and position $P = (1, 2, 3)$ with `excalc`.
Get *code*.
Get *code* for fig.8.

Exercise 13. (the gradient of a 0-form in $\mathbb{R}^2$ in polar coordinates)
Let $f : \mathbb{R}^2 \to \mathbb{R}, (r, \theta) \mapsto r - 2 \cdot \sin(\theta)$ be a 0-form, i.e. $f \in \Omega^0\mathbb{R}^2$.
a. Calculate $\text{grad}(f)$ in `excalc`.
b. Calculate $\text{grad}_P(f)$ at position $P = (1, 45^o)$ with `excalc`.

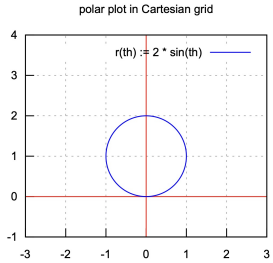


Figure 8: Polar plot of the relation $r = 2 \cdot \sin(\theta)$. The manifold M is the 0-level set $f^{-1}(0)$ of the 0-form $f(r, \theta) := r - 2 \cdot \sin(\theta)$.

Get *code*.

6 Exterior Product and eps Tensor

6.1 The wedge product aka the Exterior Product

We need a way to multiply a p-form by a q-form to obtain a $(p+q)$-form – for example, we want to be able to multiply two 1-forms to construct a 2-form. In exterior calculus, this is done using the so-called *wedge* product or *exterior product*, denoted by the symbol $\wedge$. Fortunately, we do not need to memorize an explicit difficult formula, as the exterior calculus package `excalc` handles this definitional work for us.

Fact 3. (The wedge product $\wedge$ is anti-commutative for 1-forms)

$$\begin{aligned} \wedge : \Omega^p M \times \Omega^q M &\longrightarrow \Omega^{p+q} M \\ (\omega, \nu) &\mapsto \omega \wedge \nu := ? \end{aligned} \tag{6.1}$$

with $\omega \wedge \nu = -\nu \wedge \omega$ for all $\omega, \nu \in \Omega^k M$ (anti-commutativity) [13]

Mental image
[14]

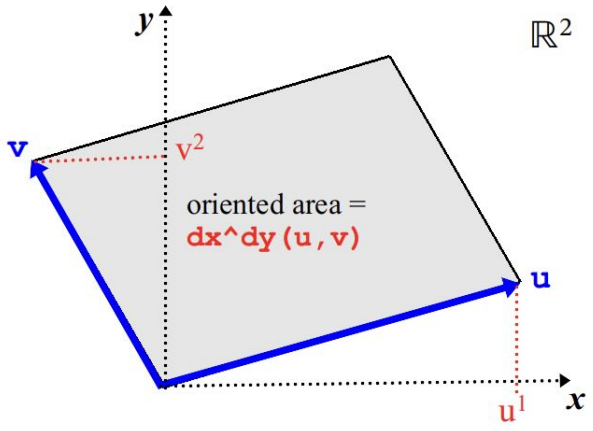


Figure 9: `dx∧dy` 'acting' on the tangent vectors **u** and **v**. That means:
The formula `dx∧dy(u,v)` is interpreted as the determinant $det\begin{bmatrix} u^1 & v^1 \\ u^2 & v^2 \end{bmatrix} = u^1v^2 - u^2v^1$ of a 2×2 matrix $M = (\vec{u}, \vec{v})$ and allows a geometrical interpretation as *oriented* area of a paralleogram spanned by the vectors $u = \binom{u_1}{u_2}$ and $v = \binom{v_1}{v_2}$,
i.e. `dx∧dy(u,v)` $\overset{p.d.}{=} det\begin{bmatrix} u^1 & v^1 \\ u^2 & v^2 \end{bmatrix} \equiv u^1v^2 - u^2v^1$.

[13] where the ? stands for a complicated formula.
[14] cf. ARNOLD [3, p.164, example 1]

Example 8. Use `excalc` to verify the following facts. Let $\omega, \nu \in \Omega^1 M$.

a. $\omega \wedge \nu = -\nu \wedge \omega$

b. $dx \wedge dx = 0$. Give an intuitive argument using fig. 7.

c. $\omega = 2dx - 3dy$ and $\nu = dx + 5dy$, calculate $\omega \wedge \nu$ and $\nu \wedge \omega$.

d. Calculate the `wedge` product of the two 1-forms $\omega = u_1 dx + u_2 dy + u_3 dz$ and $\nu = v_1 dx + v_2 dy + v_3 dz$. Guess?

Solution

```
load_package excalc;

pform omega =1, nu=1;   % ---------- a.
omega ^ nu;
 nu ^ omega;

pform {x,y}=0;          % ---------- b.
d x ^ d x;

omega := 2*d x -3*d y;  % ---------- c.
nu := d x + 5 * d y;
omega ^ nu;
nu ^ omega;

pform {x,y,z}=0;        % ---------- d.
omega := u_1*d x + u_2*d y +u_3*d z;
nu    := v_1*d x + v_2*d y +v_3*d z;
nu ^ omega;   % the result is alphabetical sorted
```

REDUCE output:

```
omega := u_1*d x + u_2*d y +u_3*d z;
```

$$\omega := d\,x\,u_{-1} + d\,y\,u_{-2} + d\,z\,u_{-3}$$

```
nu    := v_1*d x + v_2*d y +v_3*d z;
```

$$\nu := d\,x\,v_{-1} + d\,y\,v_{-2} + d\,z\,v_{-3}$$

```
nu ^ omega;
```

$$-d\,x \wedge d\,y\,u_{-1}\,v_{-2} + d\,x \wedge d\,y\,u_{-2}\,v_{-1} - d\,x \wedge d\,z\,u_{-1}\,v_{-3} + d\,x \wedge d\,z\,u_{-3}\,v_{-1} - d\,y \wedge d\,z\,u_{-2}\,v_{-3} + d\,y \wedge d\,z\,u_{-3}\,v_{-2}$$

Get *code*.

Remark. From d. we see: the components of the 2-form $\omega \wedge \nu$ are the same as the components of the cross product $\vec{\omega} \times \vec{\nu}$. We therefore may regard the wedge-product as an n-dimensional generalisation of the 3-dimensional cross product.

6.1.1 Exercises

Exercise 14. Determine the value of the 1-form $\omega = 2dx - 3dy$ on the tangent vector $v = (1, 1)$ at the point $P = (4, 5)$
a. using the action $\omega_P(v)$ analog fig.9
b. using a built-in operators of `excalc`.

Get *code*.

Exercise 15. (from ARAPURA [2, p.9])
∘ A C^1 1-form $\omega = Fdx + Gdy + Hdz$ is called *exact* if there is a C^2 function f (called a *potential*) such that $\omega = df$.
∘ A 1-form ω is called *closed* if $d\omega = 0$, or equivalently if $F_y = G_x, F_z = H_x, G_z = H_y$.

Determine which of the following 1-forms on $\mathbb{R}^2$ are exact.
Express the exact 1-forms in the form df.
(a) $3ydx + xdy$
(b) $ydx + xdy$
(c) $e^x ydx + e^x dy$
(d) $-ydx + xdy$

Check that the following 1-forms are exact, and express them as df.
(e) $dx + 2ydy + 3z^2dz$
(f) $zy\cos(xy)dx + zx\cos(xy)dy + \sin(xy)dz$

Exercise 16. (from ARAPURA [2, p.9])
It is sometimes possible to make a 1-form exact by multiplying it by a nonzero function of x and y called an *integrating factor*. In each of the non-exact 1-forms of the problem before, find an integrating factor of the form x^n, for some integer n.

Exercise 17. ((from ARAPURA [2, p.11])
Using the rules for the wedge product $\wedge$, verify
$(3dx + dy) \wedge (e^x dx + 2dy) = (6 - e^x)dx \wedge dy$.

Exercise 18. ((from ARAPURA [2, p.16])
Let $\alpha = xdx + ydy + zdz$, $\beta = zdx + xdy + ydz$ and $\gamma = xydz$ in the following problems.
1. Calculate
(a) $\alpha \wedge \beta$
(b) $\alpha \wedge \gamma$
(c) $\beta \wedge \gamma$
(d) $(\alpha + \gamma) \wedge (\alpha + \gamma)$

2. Calculate (a) $d\alpha$ (b) $d\beta$ (c) $d(\alpha + \gamma)$ (d) $d(x\alpha)$

Given $\omega = fdx + gdy + hdz$ such that $\omega \wedge dz = 0$, what can we conclude about f, g and h?

6.2 The LEVI-CEVITA eps tensor

When you define a metric using the `coframe` command, `excalc` connects `eps` to the geometry of the manifold and the *total antisymmetric* LEVI-CEVITA tensor `eps` is then available. The value of `eps` with an even permutation of the indices in a covariant position is taken to be $+1$.[15]

We show some examples of the use of `eps`.

Example 9. (evaluate standard permutations)
Permute the indices (x, y, z).

Solution.

```
load_package excalc;

% define 3D Cartesian space to establish indices x,y,z
coframe e(x) = d x, e(y) = d y, e(z) = d z;

eps(x, y, z);   % returns  1 (even permutation of x,y,z)
eps(y, x, z);   % returns -1 (odd permutation)
eps(x, x, y);   % returns  0 (repeating indices)
eps(1, 3, 2);   % eps only works w.r.t. a metric space
```

Get *code*.

Example 10. (Using `eps` with Coordinate Frames (coframe))

> Lower indices represent covariant components, and upper indices (written with a minus sign in REDUCE`excalc` syntax) represent contravariant components.[16]

Verify this behavior in the 2D polar coordinate system.

Solution.

```
load_package excalc;

% Define a 2D polar coordinate system with a metric
coframe o(r) = dr, o(phi) = r*dphi;

% Look at covariant (lower) vs contravariant (upper) indices
eps(r, phi);      % Returns 1
eps(-r, -phi);    % Returns 1 / r^2
                  % adjusted automatically by the metric determinant
```

Get *code*.

[15] For some information about `eps`, see e.g. [19, p.63 ff] or [20, p.63 ff]
[16] for an explanation, see e.g. [19, p.21,31,37 ff] or [20, p.20, 29,32,36 ff].

Example 11. (Metric Contractions e.g. in MINKOWSKI Spacetime)
In general relativity or field theory simulations, `eps` is frequently contracted with other tensors or vectors. `excalc` automatically processes index contractions when an upper and lower index match.

Verify.

```
load_package excalc;

% Define 4D Minkowski spacetime metric with diag(-1, 1, 1, 1)
coframe o(t) = d t, o(x) = d x, o(y) = d y, o(z) = d z
        with signature -1, 1, 1, 1;

% Now evaluate contractions inside tensor expressions
eps(t, x, y, z);          % Covariant volume element
eps(-t, -x, -y, -z);      % Contravariant volume element;
                          %    accounts for the -1 metric component
```

Get *code*.

Example 12. (The Determinant)

If $A = (a_{ij}) \in \mathbb{R}^{n\times n}$ is a n-by-n square matrix, the determinant is defined and calculated through the so-called LEIBNIZ formula using sigma notation,

$$\det(A) = \sum_{\sigma\in S_n} \operatorname{sgn}(\sigma) a_{1,\sigma_1} \cdots a_{n,\sigma_n}$$

Written with the `eps` tensor, the Leibniz formula is

$$\det(A) = \sum_{i_1,i_2,\ldots,i_n} \varepsilon_{i_1\cdots i_n} a_{1,i_1} \cdots a_{n,i_n}$$

where the sum is taken over all n-tuples of integers from $\{1,\ldots,n\}$.

This is a horrible formula, which no one really wants to compute this way by hand. So let's do REDUCE`excalc` the work for us.

For the special case of a 3×3 matrix A we have for $det : \mathbb{R}^{3\times 3} \to \mathbb{R}^1$

$$\det(\mathbf{A}) = \sum_{i=1}^{3}\sum_{j=1}^{3}\sum_{k=1}^{3} \varepsilon_{ijk} a_{1i} a_{2j} a_{3k} \tag{6.2}$$

Here ε_{ijk} acts as a filter, which sort out the proper ones of the whole 27 summands.

Let's realize this in REDUCE[excalc] in two different ways.
1st we use the Levi-Civita tensor `eps` to calculate the *determinant expansion* of a matrix.
2nd we define the rows/columns of the matrix as a set of linear 1-forms.

Solution 1:
We do an explicit summation loop through all permutations using `eps` to extract the full determinant expansion.

```
load_package excalc;

% 1. Define a 3D coordinate frame to register indices (x,y,z) aka (1,2,3)
coframe e(1)=d x, e(2)=d y, e(3)=d z;

% 2. Define a core Reduce matrix A and fill it with symbolic variables
matrix a(3,3);
a(1,1):=a11; a(1,2):=a12; a(1,3):=a13;
a(2,1):=a21; a(2,2):=a22; a(2,3):=a23;
a(3,1):=a31; a(3,2):=a32; a(3,3):=a33;
a;

% 3. Calculate determinant expansion using eps
det_expansion := for each i in {1,2,3} sum
                 for each j in {1,2,3} sum
                 for each k in {1,2,3} sum
                     eps(i,j,k) * a(1,i) * a(2,j) * a(3,k);

% 4. Define a matrix and fill it with concrete values
matrix M(3,3);
M(1,1):=2; M(1,2):=2; M(1,3):=4;
M(2,1):=1; M(2,2):=2; M(2,3):=-5;
M(3,1):=3; M(3,2):=1; M(3,3):=-3;
M;

det_expansion := for i:=1:3  sum
                 for j:=1:3  sum
                 for k:=1:3  sum
                     eps(i,j,k) * m(1,i) * m(2,j) * m(3,k);
```

Get *code*.

Solution 2:
Alternatively, calculate the determinant by defining the rows/columns of the matrix as 1-forms, take their exterior product $\wedge$, and contract the resulting volume form.

```
load_package excalc;

% 1. Define a 3D coordinate frame and three 1-forms
coframe e(1)=d x, e(2)=d y, e(3)=d z;
pform row1=1, row2=1, row3=1;

% 2. Define the rows of matrix A as 1-forms filled with symbolic variables
row1 := a11*e(1) + a12*e(2) + a13*e(3);
row2 := a21*e(1) + a22*e(2) + a23*e(3);
row3 := a31*e(1) + a32*e(2) + a33*e(3);

 % 3. Calculate determinant using wedge product
det_form := row1 ^ row2 ^ row3;

 % 2. Define the rows of matrix A as 1-forms filled with concrete values
row1 := 2*e(1) + 2*e(2) + 4*e(3);
row2 := 1*e(1) + 2*e(2) - 5*e(3);
row3 := 3*e(1) + 1*e(2) - 3*e(3);

 % 3. Calculate determinant value using wedge product
det_form := row1 ^ row2 ^ row3;
```

REDUCE output:

```
coframe e(1)=d x, e(2)=d y, e(3)=d z;

pform row1=1, row2=1, row3=1;

row1 := 2*e(1) + 2*e(2) + 4*e(3);
row2 := 1*e(1) + 2*e(2)  - 5*e(3);
row3 := 3*e(1) + 1*e(2)  - 3*e(3);
```

$$\mathrm{row}_1 := 2\,(e^1 + e^2 + 2\,e^3)$$

$$\mathrm{row}_2 := e^1 + 2\,e^2 - 5\,e^3$$

$$\mathrm{row}_3 := 3\,e^1 + e^2 - 3\,e^3$$

```
det_form := row1 ^ row2 ^ row3;
```

$$\mathrm{det_form} := -46\,e^1 \wedge e^2 \wedge e^3$$

- Calculation of determinant using wedge product of 1-forms.
- The result is the factor in front of the volume form $e1 \wedge e2 \wedge e3$.

Get *code*.

6.2.1 Exercises

Exercise 19. (how to take the value -46 from the determinant form in fig.12)
Try to capture the value of the determinant from the determinant form in example.12.b.
Get *code*.

Exercise 20. (definition of 3-by-3 determinant using LEIBNIZ formula)
Use a self-made permutation operator $Eijk$ and do the three sum's in (6.2) explicit.
You may look at the following EIGENMATH code:

```
A : matrix([1,2,3],[4,5,6],[7,8,8]);    /* a 3-by-3 matrix */
Eijk(i,j,k) := (j-i)*(k-i)*(k-j)/2;    /* permutation symbol as function */

Det(M) := sum(sum(sum(  Eijk(i,j,k)*M[1,i]*M[2,j]*M[3,k],      /* see (6.2) */
                                    i,1,3), j,1,3), k,1,3);
Det(A);
determinant(A);      /* check with build-in det-function */
```

Exercise 21. (4D determinant)
Do the implementation of a 4D determinant and a reconstruction of the Levi-Civita tensor by [20, p.67] in REDUCE[excalc].

Exercise 22. (cross product)
Reproduce the implementation of the `cross` product $\times$ in [20, p.71] in REDUCE[excalc].

Exercise 23. (curl)
Reproduce the implementation of the `curl` to be seen in [20, p.74] in REDUCE[excalc].[17]

[17] `cross` and `curl` are also pre-defined in REDUCE packages `Avector` and `Orthovec`. But here we want to reproduce these operators in REDUCE package `excalc`.

7 2 - Forms

The exterior derivative of a one form is a two form. You can use Hodge duality to convert it to a one form, and then the dot product to make it a vector. That's the curl. [18]

Q: What is a typical 2-form?
A: the `curl(F)` of a 1-form $F : \mathbb{R}^n \to \mathbb{R}$ or
the flux of a fluid over the area of a parallelogram; cf. [3, p.164].

Fact 4. (fundamental relations)
(6) The canonical basis of the space $\Omega^2\mathbb{R}^n$ are the 2-forms $dx^1 \wedge dx^2, ..., dx^1 \wedge dx^n$.[19]
(7) The basis 2-forms $dx \wedge dy$ etc. in pick out the v_1, v_2 components of a tangent vector v.
○ 2-forms are declared in `excalc` using the key word `pform omega=2;`.

Mental image

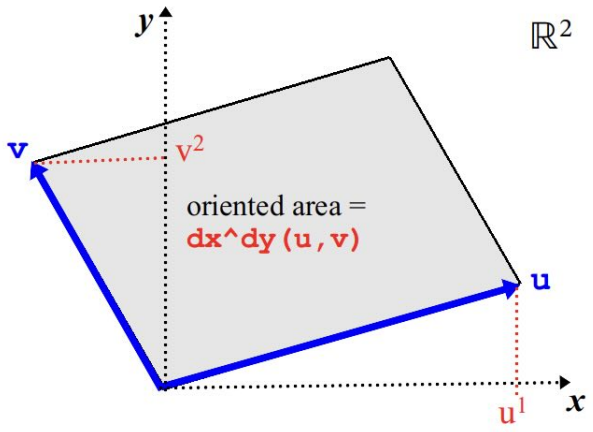


Figure 10: Remember: `dx∧dy` 'acting on the tangent vectors **u** and **v**' means:
`dx∧dy(u,v)` $:= det\begin{bmatrix} u^1 & v^1 \\ u^2 & v^2 \end{bmatrix} = u^1v^2 - u^2v^1 =$ *oriented* area of $\diamond(u, v)$.

Inspired by fig.10, we define

Definition (*action of a 2-form on 2 vectors*)**.**
The 'action' $dx \wedge dy(v, w)$ of a 2-form $\omega = dx \wedge dy$ on a pair of tangent vectors v, w is defined by the *evaluation* of v and w by ω, i.e.

$$dx \wedge dy(v, w) := det\begin{bmatrix} v \lrcorner dx & w \lrcorner dx \\ v \lrcorner dy & w \lrcorner dy \end{bmatrix} \tag{7.1}$$

[18] A. Winkler, cf. relations
[19] Remember: the superscripts $...^1, ...^2, ...$ are indices, not exponents.

Example 13. (*action of a 2-form on 2 vectors in* REDUCE[excalc], see [8]).
Suppose we have the 2-form $\omega = (x + 2z) \cdot dx \wedge dy + y \cdot dx \wedge dz$.
Calculate the action of ω on the tangent vectors $v_1 = (1, 3, 3)$ and $v_2 = (-1, 0, 7)$.
Determine the value of $\omega(v_1, v_2)$ at the point $(1, 1, 5)$. 43

Solution.

We use the identifiers xx etc. to distinguish these variables from the base forms x etc.
Using REDUCE[excalc]:

```
load_package excalc;
spacedim 3;
pform {x,y,z, xx,yy,zz}=0;
pform w =1;                % w means omega

w := (xx + 2*zz) * d x ^ d y + yy * d x^d z;

a :=  1*@ x + 3*@ y + 3*@ z;   %  i.e.  (1,3,3) in TpR^3
b := -1*@ x + 0*@ y + 7*@ z;   %        (-1,0,7)

% action i.e. evaluation  w(a,b) using (7.1)

(xx+2*zz) * det mat((a _| d x, a _| d y),
                    (b _| d x, b _| d y))
     + yy * det mat((a _| d x, a _| d z),
                    (b _| d x, b _| d z));

sub(xx=1, yy=1, zz=5, ws);  % at point (1,1,5)
```

REDUCE[excalc] output:

```
(xx+2*zz) * det mat((a _| d x, a _| d y),
                    (b _| d x, b _| d y))
     + yy * det mat((a _| d x, a _| d z),
                    (b _| d x, b _| d z));
```

$$3\,xx + 10\,yy + 6\,zz$$

```
sub(xx=1, yy=1, zz=5, ws);
```

$$43$$

Get *code*.

Example 14. (*a potpourri of calculations for* `dx∧dy(`$\begin{bmatrix}1\\2\end{bmatrix}, \begin{bmatrix}3\\4\end{bmatrix}$`)`)
Calculate the wedge procuct `dx∧dy(u,v)` for $u = \begin{bmatrix}1\\2\end{bmatrix}, v = \begin{bmatrix}3\\4\end{bmatrix}$ in different ways including the interpretation in fig.12 as the determinant of the matrix $\begin{bmatrix}1 & 2\\3 & 4\end{bmatrix}$.

a. Use a self-defined function *det*2 along fig.11.
b. Use the `matrix` operator and the built-in operator `det` of core REDUCE.
c. Use package `Avector` and the interpretation of `det` as 2-dim `cross` product with both vectors lying in the xy-plane of $\mathbb{R}^3$.
d. Use package `Orthovec` and the interpretation of `det` as 2-dim `cross` product.
e. Use directly the wedge product in package `excalc`.

Solution

```
% a. ---------- core Reduce
procedure det2(u,v); part(u,1)*part(v,2)-part(u,2)*part(v,1);
det2({1,2},{3,4});

% b. ---------- core Reduce with built-in det
matrix A(2,2);
A(1,1):=1;
A(1,2):=2;
A(2,1):=3;
A(2,2):=4;
A;
det(A);

% c. ---------- Reduce package AVECTOR
%    We interpret det as 2-dim cross product with
%    both vectors lying in the XY plane of R^3
load_package avector$

u := avec(1, 2, 0);
v := avec(3, 4, 0);
w := u cross v;          % result in 3rd component

% d. ---------- Reduce package ORTHOVEC
load_package orthovec$
u := svec(1, 2, 0);
v := svec(3, 4, 0);
u><v;                    % result in 3rd component

% e. ---------- Reduce package EXCALC
load_package excalc;
pform {x,y}=0;
spacedim 2;
u:=1*@ x + 2*@ y;
v:=3*@ x + 4*@ y;
```

```
v  _|  (u  _|  (d x ^ d y));
(3*@ x + 4*@ y) _| ((1*@ x + 2*@ y) _|  (d x ^ d y));
```

REDUCE output:

```
v _| (u _| (d x ^ d y));
                                                    - 2
(3*@ x + 4*@ y) _| ((1*@ x + 2*@ y) _| (d x ^ d y));
                                                    - 2
```

Get *code*.

7.1 Exercises

Exercise 24. (the differential of a 2-form in $\mathbb{R}^3$)
Find $d\eta$ for the 2-form $\eta = (x + z^2)\, dx \wedge dy$.
Note: you start with a 2-form and finish with a 3-form.

Get *code*.

Exercise 25. (2-forms acting on two vectors; from COLLIER [9, p.48])
Find the value of the 2-form $\omega = (x + 2y)\, dx \wedge dy \; + \; y\, dx \wedge dz$ acting on the vectors $u = (1, 3, 3)$ and $v = (-1, 0, 7)$ at the point $P = (1, 1, 5)$.

Get *code*.

Exercise 26. (from ARNOLD [3, p.178, problem 9])
Calculate the value of the forms $\omega_1 = dx_2 \wedge dx_3$, $\omega_2 = x_1\, dx_3 \wedge dx_2$ and $\omega_3 = dx_3 \wedge dr^2$ $(r^2 = x_1^2 + x_2^2 + x_3^2)$ on the pair of vectors $\xi = (1, 1, 1)$, $\eta = (1, 2, 3)$ at the point $\mathbf{x} = (2, 0, 0)$.

$\omega_1 = 1, \omega_2 = -2, \omega_3 = -8$

Get *code*.

8 3 - Forms

They[20] are all constructed from the 'exterior derivative', together with the 'dot product' and its associated 'Hodge duality'.
The exterior derivative of a two form is a three form. You can use Hodge duality to convert it into a 0 form, that is a function. That's the divergence.[21]

Q: What is a typical 3-form?
A: the `div(f)` of a 2-form f or the oriented volume of a parallelepiped, cf. [3, p.166].

Fact 5. (fundamental relations)
(8) Each $\omega \in \Omega^k\mathbb{R}^3$ is a *alternating multilinear* map.
(9) We have: $dim\ \Omega^k\mathbb{R}^n = \binom{k}{n}$.
(10) In particular: $dim\ \Omega^3\mathbb{R}^3 = \binom{3}{3} = 1$ with $dx \wedge dy \wedge dz$ as the *unique* basic 3-form.
(11) The exterior derivative of a 3-form ω in $\mathbb{R}^3$ is zero: $d\omega = 0$.
◦ 3-forms are declared in `excalc` using the key word `pform omega=3;`.

Mental image

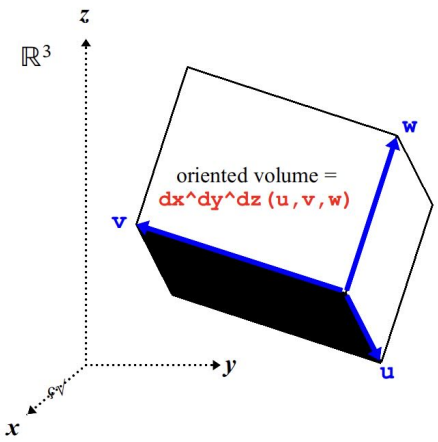


Figure 11:

The Laplace *expansion of the determinant* of a 3-by-3 matrix is

$$det\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} = a \cdot det\begin{bmatrix} e & f \\ h & i \end{bmatrix} - b \cdot det\begin{bmatrix} d & f \\ g & i \end{bmatrix} + c \cdot det\begin{bmatrix} d & e \\ g & h \end{bmatrix}$$

and allows a geometrical interpretation as *oriented* volume of a parallelotop.
Therefore `dx∧dy∧dy` 'acting on the tangent vectors **u**, **v**, **w**' means:
use the formula `dx∧dy∧dy`(u,v,w) $\equiv \det(u, v, w)$.

[20] i.e. gradient-curl-divergence-Laplacian
[21] A. Winkler, cf. relations

- 3-forms in $\mathbb{R}^3$ are of the type $\omega = f(x,y,z) \cdot dx \wedge dy \wedge dz$.

Example 15. (*a potpourri of calculations for* `dx∧dy∧dz`(u,v,w) $\equiv \det(u,v,w)$)
Calculate the wedge procuct `dx∧dy∧dy`(u,v,w) for $u = (2,2,4)^t, v = (1,2,-5)^t$ and $w = (3,1,-3)^t$ in different ways including the interpretation in fig.11 as the determinant of the matrix (u,v,w).

a. Use a self-defined operator *det*3 along fig.12.[22]
b. Use the `matrix` operator and the built-in operator `det` in $\mathbb{R}^3$.
c. Use package `Avector` and the interpretation of `det` as `spat`[23] aka `Box-product`.
d. Use package `Orthovec` and the interpretation of `det` as `spat` volume.
e. Use directly the wedge product in package `excalc`.

Solution

```
% a. ---------- core Reduce with self-made operator det
operator e;
noncom e;
for all j let e(j)*e(j)=0;
for all j,k such that (j>k) let e(j)*e(k)=-e(k)*e(j);

%         u=(2,2,4)  v=(1,2,-5)  w=(3,1,-3)
det3 := ( 2*e(1)+2*e(2)+4*e(3) ) *
        ( 1*e(1)+2*e(2)-5*e(3) ) *
        ( 3*e(1)+1*e(2)-3*e(3) );
% where e(1)e(2)e(3)= 1 = vol(unit box)

% b. ---------- core Reduce with built-in det.
%               u=(2,2,4)  v=(1,2,-5)  w=(3,1,-3)
matrix M(3,3);
M(1,1):=2; M(1,2):=2;  M(1,3):=4;
M(2,1):=1; M(2,2):=2;  M(2,3):=-5;
M(3,1):=3; M(3,2):=1;  M(3,3):=-3;
M;
det(M);

% c. ---------- Reduce package AVECTOR
%    We interpret det as 3-dim spat product in R^3
load_package avector$

u := avec(2,2,4);
v := avec(1,2,-5);
w := avec(3,1,-3);
det := (u cross v) dot w;
```

[22] Code inspired by [47, p.35]
[23] cf. [23, p.61]

```
% d. ---------- Reduce package ORTHOVEC
load_package ORTHOVEC$

u := svec(2,2,4);
v := svec(1,2,-5);
w := svec(3,1,-3);
det := (u><v) * w;

% e. ---------- Reduce package EXCALC
load_package excalc;
pform {x,y,z}=0;
spacedim 3;
u := 2*@ x + 2*@ y + 4*@ z;          % u = (2,2, 4)
v := 1*@ x + 2*@ y - 5*@ z;          % v = (1,2,-5)
w := 3*@ x + 1*@ y - 3*@ z;          % w = (3,1,-3)
w _| (v  _|  (u  _|  (d x ^ d y ^ d z)));
```

REDUCE output for e. :

```
      % u=(2,2,4)
v := 1*@ x + 2*@ y - 5*@ z;
```

$$v := \partial_x + 2\,\partial_y - 5\,\partial_z$$

```
      % v=(1,2,-5)
w := 3*@ x + 1*@ y - 3*@ z;
```

$$w := 3\,\partial_x + \partial_y - 3\,\partial_z$$

```
      % w=(3,1,-3)
w _| (v  _|  (u  _|  (d x ^ d y ^ d z)));
```

$$-46$$

Get *code*.

Inspired by the last example part e., we define

Definition (*action of a 3-form on 3 vectors*)**.**
The 'action' $\omega(u,v,w)$ of a 3-form ω on a triple of tangent vectors u,v,w is defined by the *evaluation* formula

$$\omega(u,v,w) := w \lrcorner (v \lrcorner (u \lrcorner \omega))) \tag{8.1}$$

Remark. You may define an analogous formula to (7.1) as equivalent to formula (8.1). This procedure is then easily generalizable to more than 3 dimensions.

8.1 Exercises

Exercise 27 (3-forms acting on three vectors, from [15])**.**
Find the value of the basis 3-form $\omega = dx_1 \wedge dx_2 \wedge dx_4$ acting on the tangent vectors $v_1 = (1, 2, -1, 1)$, $v_2 = (3, -2, 1, 2)$ and $v_3 = (0, 1, 2, 1)$ of $Tp\mathbb{R}^4$. -7

Exercise 28. (evaluation function for the wedge product `dx∧dy∧dy`(u,v,w))

> The 'action' of the wedge product on the tangent vectors **u, v** and **w** is noted in the books by COLLIER [9] and BACHMAN [4] by the term `dx∧dy∧dy (u,v,w)`, which is interpreted as a determinant along fig.12 and allows a geometrical interpretation as *oriented* area of a parallepiped spanned by the three vectors u, v and v.

Program a REDUCE function `dxdydz(u,v,w)` to realize this evaluation for 3-elements lists u, v, w and verify it for $u := \{2, 2, 4\}, v := \{1, 2, -5\}$ and $w := \{3, 1, -3\}$.

Hint: use package `listvecops`.

Exercise 29. (from ARNOLD [3, p.178, problem 12])
Let x_1, x_2 and x_3 be local coordinates in the euclidean three-space $\mathbb{R}^3$. Then the square of the *line element* has the form

$$ds^2 = E_1 dx_1^2 + E_2 dx_2^2 + E_3 dx_3^2 \tag{8.2}$$

Find E_1, E_2, E_3 for cartesian coordinates x, y, z, for cylindrical coordinates r, φ, z and for spherical coordinates $R, \varphi.\theta$ in the euclidian space $\mathbb{R}^3$.

$ds^2 = dx2 + dy^2 + dz^2 = dr^2 + r^2 d\varphi^2 + dz^2 = dR^2 + R^2 cos^2\theta d\varphi^2 + R^2 d\theta^2$

9 HODGE ⋆ operator

How are a gradient, curl, divergence, Laplacian, etc. related to each other? They are all constructed from the 'exterior derivative', together with the 'dot product' and its associated 'HODGE duality'. [24]

The HODGE star operator (usually denoted by $\star$) is a fundamental linear map in differential geometry. On an n-dimensional oriented Riemannian manifold like $\mathbb{R}^n$, it relates the spaces of k-forms to the spaces of $(n-k)$-forms.
Here is a summary of the HODGE $\star$ operator's key mathematical properties.

Fact 6. (*The* HODGE $\star$ *dual is a linear map with remarkable characteristics)*[25]

$$\begin{array}{ccc} \star : \Omega^k\mathbb{R}^n & \longrightarrow & \Omega^{n-k}\mathbb{R}^n \\ \omega & \mapsto & \star\omega := ? \end{array} \tag{9.1}$$

(1) Linearity: the HODGE-$\star$ is a linear map. For two k-forms α, β and $a, b \in \mathbb{R}$ we have:

$$\star(a\alpha + b\beta) = a(\star\alpha) + b(\star\beta) \tag{9.2}$$

(2) Degree shift: $\star$ maps a k-form to a $(n-k)$-form, i.e. (9.1)

(3) Involution: Applying the HODGE-$\star$ twice in succession to a k-form ω yields the original form, up to a sign. This sign depends on the dimension of the space n, the degree of the form k, and the signature of the metric s (the number of negative signs):

$$\star\star\,\omega = (-1)^{k(n-k)+s}\omega \tag{9.3}$$

(4) Duality of scalars and volume elements:
1. The HODGE star of a function (0-form) yields the volume element `vol` (the n-form in $\mathbb{R}^n$):

$$\star\,1 = \mathrm{vol} \tag{9.4}$$

2. For the volume element (taking into account the signature s), we have:

$$\star\,\mathrm{vol} = (-1)^s \cdot 1 \tag{9.5}$$

(5) Relationship to the inner product: The HODGE $\star$ is closely related to the inner product $\langle\cdot,\cdot\rangle$ of differential forms. For any two k-forms α, β , the following holds:

$$\alpha \wedge \star\beta = \langle\alpha, \beta\rangle \cdot \mathrm{vol} \tag{9.6}$$

This means that the wedge product (exterior product) of a form and the Hodge dual of another form directly yields their scalar product multiplied by the volume element.

Remark. 1. In Euclidean spaces ($s = 0$): the factor in (9.3) simplifies to $(-1)^{k(n-k)}$.
2. In 3D space (n=3): for a 1-form (k=1) we have $\star\star\,\omega = (-1)^{1\cdot 2}\omega = \omega$.

[24] A. Winkler, cf. relations
[25] where the ? stands for a complicated formula.

Example 16. (The HODGE star ⋆ operator is noted # in REDUCE[excalc])

$$\star 1 = dx \wedge dy$$
$$\star dx = dy$$
$$\star dy = -dx$$
$$\star (dx \wedge dy) = 1$$

HODGE star ⋆ in $\mathbb{R}^2$

Verify these propositions of the mathematical HODGE star ⋆ in REDUCE[excalc] with #.

Solution:

```
load_package excalc;
spacedim 2;        % we are in R^2

coframe e(x) = d x, e(y) = d y with metric g = e(x)*e(x) + e(y)*e(y);
# 1;               % = dx ^ dy      (1)
# d x;             % = dy
# d y;             % = - dx
# (d x ^ d y);     % = 1            (2)
```

Get *code*.

- Because of property (1) and (2) $dx \wedge dy$ resp. 1 is called the *volume* (area) form of $\mathbb{R}^2$.

Example 17. (Properties of the HODGE star ⋆ operator in REDUCE[excalc])
Verify, that the propositions of the mathematical HODGE star ⋆ in Fact 6, (1) .. (5) are realized by the Hodge-star dual # in REDUCE[excalc] in $\mathbb{R}^3$.

Solution:

```
load_package excalc;
spacedim 3;
coframe e(1) = d x, e(2) = d y, e(3) = d z;
pform alpha=1, beta=1;

linearity := # (a*alpha + b*beta) - (a*(# alpha) + b*(# beta));

exdegree # alpha;

volume_form := # 1;

volume_star := # (e(1)^e(2)^e(3));

# # alpha;
```

Get *code*.

Example 18. (the `cross` × product in $\mathbb{R}^3$ – a mini project)
Given two linearly independent vectors a and b in $\mathbb{R}^3$, the *cross product* $a \times b$ is a vector, that is perpendicular to both a and b, i.e. therefore perpendicular to the plane containing them. Using column vector notation, we have the explicit defining formula for $a \times b$:

$$\begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} \times \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} := \begin{bmatrix} a_2b_3 - a_3b_2 \\ a_3b_1 - a_1b_3 \\ a_1b_2 - a_2b_1 \end{bmatrix} \tag{9.7}$$

In any basis, the cross-product $a \times b$ is given by the tensorial formula

$$(a \times b)_k = \sum_{i=1}^{3} \sum_{j=1}^{3} \varepsilon_{ijk} a^i b^j \tag{9.8}$$

Therein ε_{ijk} is the permutation symbol, which respects the positions of the indices.

Task: Define the `cross` product for two vectors $a = (a_1, a_2, a_3), b = (b_1, b_2, b_3) \in \mathbb{R}^3$ and check it for the data $a = (1, 0, 2x)$ and $b = (0, 1, 2)$ by

a. using the REDUCE package `Avector`
b. using the REDUCE package `Orthovec`
c. using the REDUCE package `listvecops`
d. using the REDUCE package `excalc` and differential forms
e. using a self-defined function `Eijk(i,j,k)` for implementing (9.8).

Solution

ad a.: `cross` is defined in package `avector` as an infix operator:

```
load_package Avector$

a := avec(a_1,a_2,a_3);
b := avec(b_1,b_2,b_3);
a cross b;

a := avec(1, 0, 2*x);
b := avec(0, 1, 2);
a cross b;
```

Get *code*.

ad b.: `cross` is defined in package `orthovec` as an infix operator noted '`><`':

```
load_package ORTHOVEC$

a := svec(a_1,a_2,a_3);
b := svec(b_1,b_2,b_3);
a  >< b;
```

```
a := svec(1, 0, 2*x);
b := svec(0, 1, 2);
a ><  b;
```

Get *code*.

ad c.: In package `listvectops` we use lists to handle vectors. For accessing list (vector) elements, the new operator `_ (lnth)` can be used instead of the `part` operator of core REDUCE. Note that the infix operator `_` must be separated from the name of the list variable to which it is applied. Also, one has to use the declaration `listproc` instead of the reserved name `procedure` to program a function.

```
load_package LISTVECOPS$

% function CROSS in package listvecops using (9.7)
listproc cross(u,v);
   begin scalar x,y,z, Cx,Cy,Cz;
     Cx := u _(2)*v _(3) - u _(3)*v _(2);
     Cy := u _(3)*v _(1) - u _(1)*v _(3);
     Cz := u _(1)*v _(2) - u _(2)*v _(1);
     return {Cx,Cy,Cz}
   end;

a := {a_1,a_2,a_3};
b := {b_1,b_2,b_3};
cross (a,b);

a := {1, 0, 2*x};
b := {0, 1, 2};
cross (a,b);
```

Get *code*.

ad d.: In REDUCE$^{\texttt{excalc}}$, the vector cross product is *implemented using the Hodge star operator* `#` and the exterior wedge product ($\wedge$). By translating 3D vectors into differential 1-forms, the cross product $\vec{a} \times \vec{b}$ corresponds to the nice and short operation

$$\mathbf{a} \times \mathbf{b} := \#(\mathbf{a} \wedge \mathbf{b}) \tag{9.9}$$

```
load_package excalc;

% Euclidean 3-space with orthonormal coframe ex,ey,ez
% metric is necessary for using HODGE star operator #
```

```
coframe e(x)=d x, e(y)=d y, e(z)=d z with signature 1,1,1;
pform {x,y,z} = 0;

% cross product on 1-forms
procedure cross(a,b);  # (a ^ b);

% check
pform {a1,a2,a3, b1,b2,b3} = 0;
pform {a,b,c} = 1;

a := a1*d x + a2*d y + a3*d z;
b := b1*d x + b2*d y + b3*d z;
c := cross( a,b );

a := 1*d x + 0*d y + 2*x*d z;
b := 0*d x + 1*d y + 2*d z;     % i.e. vec (0, 1, 2)
cross(a,b);
```

Get *code*.

ad e.: We implement *cross* in pure REDUCE as a function which uses a permutation procedure `Eijk()` in (9.8). We use 1-by-3-matrices as data type, because it seems more handy as 3-1-matrices (needs more parenthesis) or arrays: only inside the procedure *cross* we have to write $a(1, j)$ instead of $a(j)$.

```
% permutation symbol as function
procedure Eijk(i,j,k);  (j-i)*(k-i)*(k-j)/2;
Eijk(1,2,3);

procedure cross(a,b);
begin matrix c(1,3);
  for each i in {1,2,3} do
    c(1,i):= for each j in {1,2,3} sum
             for each k in {1,2,3} sum
                 Eijk(i,j, k) * a(1,j) * b(1,k);
    return c
  end;

a := mat((a1, a2, a3));
b := mat((b1, b2, b3));
cross(a,b);

a := mat((1, 0, 2*x));
b := mat((0, 1, 2));
cross(a,b);
```

REDUCE output: `(-2x -2 1)`

Get *code*.

9.1 Exercises

Exercise 30. (The HODGE star ⋆ operator in $\mathbb{R}^3$)
Check example.16 for the HODGE star ⋆ operator in $\mathbb{R}^3$.
Get *code*.

Exercise 31. (the `curl` operator in $\mathbb{R}^3$ – a mini project)
curl F [26] is a vector operator that is used in vector calculus to describe the circulation of a 3D differentiable vector field $F\colon \mathbb{R}^3 \to \mathbb{R}^3$, given through its three component functions $F(x,y,z) := (F_x(x,y,z), F_y(x,y,z), F_z(x,y,z))$.
The definition is:

$$(curl\, F)(x,y,z) := \left(\frac{\partial F_z}{\partial y} - \frac{\partial F_y}{\partial z}, \frac{\partial F_x}{\partial z} - \frac{\partial F_z}{\partial x}, \frac{\partial F_y}{\partial x} - \frac{\partial F_x}{\partial y}\right) \tag{9.10}$$

The equation for each component $(curl\, F)_k$ is obtained by exchanging each occurrence of a subscript y, z, x in cyclic permutation $(y,z) \to (z,x) \to (x,y)$ by focussing at the denominators.

Task: Implement the `curl`[27] of a vector field $F = (F_1, F_2, F_3)$ and check it for the vector field $F = (xy, -\sin(z), 1)$ by

a. using the REDUCE package `avector`
hint: `F := avec(x*y,-sin(z),1); curl F;`

b. using the REDUCE package `orthovec`
hint: example.18.c `F:= svec(x*y,-sin(z),1); curl(F);`

c. using the REDUCE package `listvecops`
hint: example.18.d and `curl(F); .. Cx := df(F[3],y)-df(F[2],z), ..`

d. using the REDUCE package `excalc` and differential forms
hint: `procedure curl(F); # d F;`
View *code*, run REDUCE.

e. using the self-defined function `Eijk(i,j,k)`.
hint: example.18.e and `Eijk(i,j,k) * df(F[k],x[j])`

[26] In classic scientific literature in Europe, the alternative notation *rot* F (read: rotation of F) is used.
[27] you may orient yourself on the mini project on *cross* in example.18

10 grad, curl, div, laplace in Exterior Calculus

We quote from the preface: A more advanced description of the well-known differential operators `grad-div-curl` can be provided using differential forms and the exterior derivative d, as Andrew WINKLER gives so aptly in response to the question
How are a gradient, curl, divergence, Laplacian, etc. related to each other?[28] on QUORA:

> They are all constructed from the 'exterior derivative', together with the 'dot product' and its associated 'Hodge duality'.
>
> The exterior derivative of a function is a one form. You can use the dot product to convert that into a vector. That's the gradient.
>
> The exterior derivative of a one form is a two form. You can use Hodge duality to convert it to a one form, and then the dot product to make it a vector. That's the curl.
>
> The exterior derivative of a two form is a three form. You can use Hodge duality to convert it into a 0 form, that is a function. That's the divergence.
>
> In three dimensions the exterior derivative of a three form is 0.

We will verify this description in this section.

Example 19. (gradient and 0-forms)
Verify, that the gradient vector field $grad(f) := \nabla f := (\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}) = \frac{\partial f}{\partial x}\cdot\mathbf{e}_1 + \frac{\partial f}{\partial y}\cdot\mathbf{e}_2 + \frac{\partial f}{\partial z}\cdot\mathbf{e}_3$ of a scalar field $f(x,y,z)\colon \mathbb{R}^3 \to \mathbb{R}$ can be associated with the exterior derivative of the 0-form $f(x,y,z)$, i.e. `grad f := d f`.

Solution

```
load_package excalc;
spacedim 3;                    % set dimension of the underlying space
coframe e(1) = d x, e(2) = d y, e(3) = d z;

pform f=0;                     % f is a scalar function
fdomain f = f(x,y,z);          % f is a function of the variables x,y and z

d f;                           % total differential of f, i.e. the gradient
```

REDUCE output:

```
fdomain f = f(x,y,z);
        % f is a function of the variables x,y and z
d f;
```

$$e^1\frac{\partial f}{\partial x} + e^2\frac{\partial f}{\partial y} + e^3\frac{\partial f}{\partial z}$$

```
                % total differential of f, i.e. the gradient
```

Get *code*.

[28] A. Winkler, cf. relations

Example 20. (curl and 1-forms)
Verify, that the curl

$$\begin{aligned} curl(F) &:= \nabla \times F := (\frac{\partial f_3}{\partial y} - \frac{\partial f_2}{\partial z}, \frac{\partial f_1}{\partial z} - \frac{\partial f_3}{\partial x}, \frac{\partial f_2}{\partial x} - \frac{\partial f_1}{\partial y}) \in \mathbb{R}^3 \\ &= (\frac{\partial f_3}{\partial y} - \frac{\partial f_2}{\partial z}) \cdot \mathbf{e}_1 + (\frac{\partial f_1}{\partial z} - \frac{\partial f_3}{\partial x}) \cdot \mathbf{e}_2 + (\frac{\partial f_2}{\partial x} - \frac{\partial f_1}{\partial y}) \cdot \mathbf{e}_3 \end{aligned} \tag{10.1}$$

of a vector field $F(x, y, z)$: $\mathbb{R}^3 \to \mathbb{R}^3$ with $F = (f_1, f_2, f_3)$ can be associated with the exterior derivative of the 1-form $F(x, y, z)$, i.e. `curl F := # d F`.

Solution

```
load_package excalc;
spacedim 3;
coframe e(1) = d x, e(2) = d y, e(3) = d z;
pform {f1,f2,f3} = 0, F = 1;      % F = om(ega) is a 1-form

fdomain f1 = f1(x,y,z);           % f1 depends on the variables x,y,z
fdomain f2 = f2(x,y,z);
fdomain f3 = f3(x,y,z);

F:= f1*d x + f2*d y+f3*d z;       % F as linear combination of dx,dy,dz
d F;                              % read of curl components of this 2-form
# d F;                            % express curl in the basis e() as (10.1)
```

REDUCE output:

```
om:= f1*d x + f2*d y+f3*d z;
```

$$\text{om} := e^1 f_1 + e^2 f_2 + e^3 f_3$$

```
d om;
```

$$-\frac{\partial f_1}{\partial y} e^1 \wedge e^2 - \frac{\partial f_1}{\partial z} e^1 \wedge e^3 + \frac{\partial f_2}{\partial x} e^1 \wedge e^2 - \frac{\partial f_2}{\partial z} e^2 \wedge e^3 + \frac{\partial f_3}{\partial x} e^1 \wedge e^3 + \frac{\partial f_3}{\partial y} e^2 \wedge e^3$$

Get *code*.

Comment. We see, that the defining term (10.1) of curl is in perfect harmony with the REDUCE$^{\texttt{excalc}}$ output.

Example 21. (divergence and 2-forms)
Verify, that the divergence

$$div(F) := \nabla \bullet F := \frac{\partial f_1}{\partial x} + \frac{\partial f_2}{\partial y} + \frac{\partial f_3}{\partial z} \in \mathbb{R} \tag{10.2}$$

of the vector field $F(x,y,z)$: $\mathbb{R}^3 \to \mathbb{R}^3$ with $F = (f_1, f_2, f_3)$ can be associated with the exterior derivative of the 2-form $F(x,y,z)$, i.e. `div F := d F`.

Solution

```
load_package excalc;
spacedim 3;
coframe e(1) = d x, e(2) = d y, e(3) = d z;
pform {f1,f2,f3} = 0, om2 = 2;

fdomain f1 = f1(x,y,z);
fdomain f2 = f2(x,y,z);
fdomain f3 = f3(x,y,z);

om2:= f1*d y ^ d z  + f2*d z ^ d x +f3*d x ^ d y;
d om2;                          % read off div from this 3-form
# d om2;                        % express div as 0-form for formula (11.2)
```

REDUCE output:

```
om2:= f1*d y ^ d z  + f2*d z ^ d x +f3*d x ^ d y;
```

$$\mathrm{om2} := e^1 \wedge e^2 f_3 - e^1 \wedge e^3 f_2 + e^2 \wedge e^3 f_1$$

```
d om2;
```

$$e^1 \wedge e^2 \wedge e^3 \left(\frac{\partial f_1}{\partial x} + \frac{\partial f_2}{\partial y} + \frac{\partial f_3}{\partial z} \right)$$

Get *code*.

Example 22. (laplacian and 3-forms)

> *How are a gradient, curl, divergence, Laplacian, etc. related to each other?* They are all constructed from the 'exterior derivative' ... and its associated 'HODGE duality'. [29]

Verify, that the laplacian[30],

$$lap(f) := \nabla^2 f := \Delta f := \frac{\partial^2 f}{\partial^2 x} + \frac{\partial^2 f}{\partial^2 y} + \frac{\partial^2 f}{\partial^2 z} \in \mathbb{R} \tag{10.3}$$

of a scalar function $f(x, y, z) : \mathbb{R}^3 \to \mathbb{R}$ can be associated with the 0-form

```
lap f = # d # d f
```  (10.4)

For a vector field $F := (f_1, f_2, f_3) : \mathbb{R}^3 \to \mathbb{R}^3$ we define $\nabla^2 F := (\nabla^2 f_1, \nabla^2 f_2, \nabla^2 f_3)$.

a. Define the laplacian of f resp. F using core REDUCE and package `avector`.
b. Define the laplacian of f resp. F using differential forms and REDUCE`excalc`.

In both tasks, check your definitions with the fields aka functions

$$f(x, y, z) := x^2 + 3y^3 : \mathbb{R}^3 \to \mathbb{R} \quad \text{resp.} \quad F := (x, x^2, x^3) : \mathbb{R}^3 \to \mathbb{R}^3$$

- Another alias name for the laplacian $lap(F)$ is `delsq(f)`, which reminds at the symbol $\nabla^2 f$ and its pronunciation of 'del squared'.

Solution

ad a.:
In core REDUCE:

```
procedure lap(f); df(f, x,2) + df(f, y,2) +df(f, z,2);
f := x^2+3*y^3;             % scalar field on R^3 to R
lap(f);

procedure vlap(f); {lap(part(f,1)), lap(part(f,2)), lap(part(f,3))};
vF := {x,x^2,x^3};          % vector field on R^3 to R^3
vlap(vF);
```

$\Delta f = 18y + 2 \quad$ resp. $\quad \Delta F = \{0, 2, 6x\}$

[29] A. Winkler, cf. relations
[30] see WIKIPEDIA *Laplacian*

With package `avector`:

```
load_package avector$

f := x^2+3*y^3;
delsq(f);

vF := avec(x,x^2,x^3);
delsq(vF);
```

Get *code*.

ad b.: We use differential forms with REDUCE[excalc].
1st we do a mathematical derivation to learn to cope with differential forms.

$$
\begin{aligned}
\star\, d \star d\,(x^2+3y^3) &\overset{1}{=} \star\, d \star \left(\frac{\partial(x^2+3y^3)}{\partial x}dx + \frac{\partial(x^2+3y^3)}{\partial y}dy\right)\\
&= \star\, d \star (2x\,dx + 9y^2\,dy)\\
&\overset{2}{=} \star\, d\,(2x\,dy - 9y^2\,dx)\\
&\overset{3}{=} \star\,(2\,dx\wedge dy - 18y\,dy\wedge dx)\\
&\overset{4}{=} \star\,(2+18y)\,dx\wedge dy\\
&\overset{5}{=} 2+18y
\end{aligned}
$$

1 : first exterior differentation of exterior form by operator d
2 : first Hodge dualization `# d f` in 2D-space
... with canonical orientation we have $\star dx = dy$ and $\star dy = -dx$
3 : second exterior differentation `d#df` where $dx\wedge dx = 0$ and $dy\wedge dy = 0$
4 : wedge rule in $\mathbb{R}^2$, see (6.1)
5 : $\star(dx\wedge dy) = 1$, because it is the volume/area form in $\mathbb{R}^2$

2nd we do the calculation of 1st using REDUCE[excalc].

```
%%---------- SCALAR LAPLACIAN
load_package excalc;

% define 3D-space with coordinates x, y, z and Euclidian metric
coframe e x = d x, e y = d y;

% declaration of function f as 0-form
pform f=0;

% definition of scalar function term
f := x**2 + 3*y**3;
```

```
% calculation step-by-step

d f;
# d f;
d # d f;
# d # d f;

% calculation of Laplace-operator (* d * d f) in one step
lap := # d # d f;

write "Laplacian(f) = ", lap;

%%---------- VECTOR LAPLACIAN
load_package excalc;

% declare a 1-dim space with coordinate x and Euclidian metric
% because F : R --> R^3
coframe e x = d x;

% declaration of the components of F as scalar 0-forms
pform f1=0, f2=0, f3=0;

% definition of the components of F=(x, x^2, x^3)
f1 := x;
f2 := x**2;
f3 := x**3;

% calculate the Laplace-operator for each component
lap1 := # d # d f1;
lap2 := # d # d f2;
lap3 := # d # d f3;

% output result
write "vectorLaplacia(F) = ", mat((lap1,lap2,lap3));
```

REDUCE output:

```
% output of result
write "vectorLaplacian(F) = ", mat((lap1,lap2,lap3));
```

$$\text{vectorLaplacian(F)} = \begin{pmatrix} 0 & 2 & 6x \end{pmatrix}$$

Get *code*.

Remark. 1. In all cases the vector calulus operators `grad-curl-div` are simple catched by the *same* expression in REDUCE[excalc]: $df - d\omega_1 - d\omega_2$ or better $df - \#d\omega_1 - \#d\omega_2$.

2. Of course, one can, for example, calculate gradient, divergence or Laplacian easily directly from the definition using partial derivatives in REDUCE[excalc] – as it could be done, for instance, in Example 22: `divF := @(f1,x) + @(f2,y) + @(f3,z);`.

3. Summary:

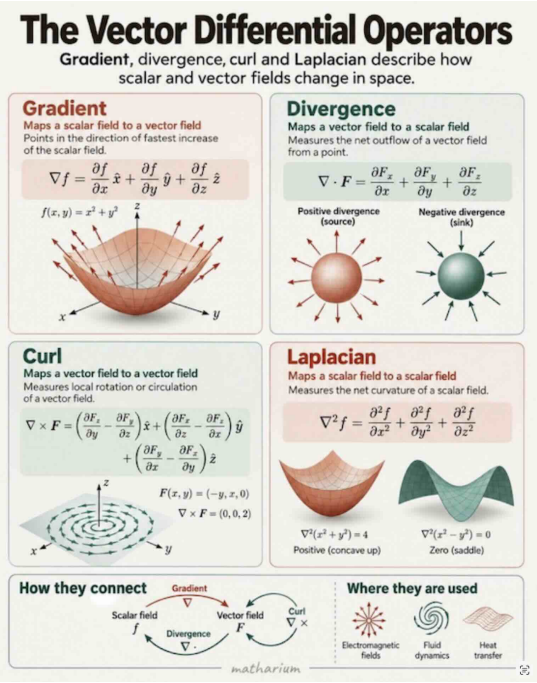
Vector Operations $\mapsto$ Differential Forms

Vector Analysis	Operation	in `excalc`	Math Notation	Output Type
Dot Product:	$\vec{a} \cdot \vec{b}$	`a _| b`	$i_{\vec{a}}\mathbf{b} =< \mathbf{a}, \mathbf{b} >$	0-form (Scalar)
Cross Product:	$\vec{a} \times \vec{b}$	`# (a ∧ b)`	$*(\mathbf{a} \wedge \mathbf{b})$	1-form (Vector)
Gradient:	∇f	`d f`	df	1-form (Vector)
Curl:	$\nabla \times \vec{F}$	`# d F`	$*d\mathbf{F}$	1-form (Vector)
Divergence:	$\nabla \cdot \vec{F}$	`d F`	$d\,\mathbf{F}$	0-form (Scalar)
Laplacian:	$\nabla^2 f := \Delta f$	`# d # f`	$*\,d*\mathbf{f}$	0-form (Scalar)

4. ARNOLD [3] gives a compact overview on p.194–196:

> In a three-dimensional oriented riemannian space M, every vector field $\mathbf{A}$ corresponds to a 1-form $\omega^1_{\mathbf{A}}$ and a 2-form $\omega^2_{\mathbf{A}}$. Therefore , exterior differentiation can be considered as an operation on vectors. Exterior differentiation of 0-forms (functions), 1-forms, and 2-forms correspond to the operations of gradient, curl, and divergence [...].

5. *matharium* at $\mathcal{X}$ gives this summary:



Laplacian Exercises

Exercise 32. (Laplacian, from [31, p.774, 53-55])
Compute all mixed partial derivates $f_{xy} := \frac{\partial^2 f}{\partial x \partial y}$ etc. and the Laplacian of
a. $u = 2xy/(x^2 + y^2)$
b. $z = \cos(xy^2)$
c. $f = e^{-xy^2} + y^3x^4$

Get *code*.

Exercise 33. (Laplace' equation, from Problem 150)
Show that $f(x, y) = \ln(x^2 + y^2)$ solves Laplace's equation $f_{xx} + f_{yy} = 0$ in $\mathbb{R}^2$.
Get *code*.

Exercise 34. (Laplace' equation, from [31, p.796, 73b]]
Show that $g(x, y, z) = \frac{1}{(x^2+y^2+z^2)^{1/2}}$ solves Laplace's equation $g_{xx} + g_{yy} + g_{zz} = 0$ in $\mathbb{R}^3$.
Get *code*.

Exercise 35. (heat equation)
The equation for heat flow in the xy-plane is $f_t = f_{xx} + f_{yy}$.
Show that $f(x, y, t) = e - 2t \sin x \sin y$ is a solution.
Get *code*.

10.1 Exercises

We now solve some reference problems from MARSDEN [31] and DINEEN [11].
We calculate in the Euclidean space $\mathbb{R}^3$.

Exercise 36. (from [31, p.797])
a. Find ∇f if $f(x,y,z) = xy - z^2$. Use REDUCE$^{\texttt{excalc}}$.
b. Find ∇f if $f(x,y,z) = \exp(xy) - x\cos(yz^2)$. Use REDUCE$^{\texttt{excalc}}$.
What is the value at point $(1,2,3)$?
c. Let $\vec{r} = x\vec{i} + y\vec{j} + z\vec{k}$ and $r := \|\vec{r}\| := \sqrt{x^2+y^2+z^2}$.
Show that $\nabla(\frac{1}{r}) = -\frac{\vec{r}}{r^3}$.
Get *code*.

Exercise 37. (from [11, p.63])
Compute the curl of the following vector fields using REDUCE$^{\texttt{excalc}}$.
a. $F(x,y,z) = \frac{(1,2,3)}{\|X\|}$.
b. $xy\vec{i} - \sin z\vec{j} + \vec{k}$. – What is the value at point $(1,2,3)$?
c. $F(x,y,z) = e^z\vec{i} - \cos(xy)\vec{j} + z^3y\vec{k}$.
Get *code*.

We now seek to identify those vector fields F in $\mathbb{R}^n$ which are the gradient of a scalar-valued function f. Vector fields of this kind are called *conservative* and are said to *have a scalar potential*, cf. [11, p.56 ff].

Definition. If $\nabla f = F$ we call f a *scalar potential* of F.

Fact 7. (existence of a potential, [11, p.56 ff])
If $F = (f_1, ..., f_n) : U \to \mathbb{R}^n$ is a continuously differentiable vector field on a suitable open set U in $\mathbb{R}^n$, then: F has a scalar potential $\Leftrightarrow \frac{\partial f_i}{\partial x_j} = \frac{\partial f_i}{\partial x_j}, \forall i,j$.

Exercise 38. (from [11, p.59])
a. Verify that $F(x,y,z) = (ye^z, xez, xye^z)$ has a potential. Use REDUCE$^{\texttt{excalc}}$.
b. Find a potential for the vector filed F in a. Use REDUCE$^{\texttt{excalc}}$.
c. Check the existence and find a scalar potential for the vector fields .
$F(x,y,z) = (xy, yz, zx)$ and $G(x,y,z) = (y, \cos yz, y\cos yz)$.
Get *code*.

Exercise 39. (from [31, p.932, P.21])
a. Check if the vector field $\Phi(x,y) = (3x^2y^2, 2x^3y^2)$ is conservative;
if it is, find an f such that $\Phi = \nabla f$ Use REDUCE$^{\texttt{excalc}}$.
b. Check if the vector field $\Phi(x,y) = 2xy\mathbf{i} + (3z^3 + x^2)\mathbf{j} + 9yz^2\mathbf{k}$ is conservative;
if so, find function f such that $\Phi = \nabla f$ Use REDUCE$^{\texttt{excalc}}$.
Get *code*.

Exercise 40. (from [31, p.930])
Compute the divergence of the following vector fields using REDUCE$^{\texttt{excalc}}$.
a. $\Phi(x,y) := (x^3, -x \cdot \sin(xy)$. What is the value at point $(1,2)$?
b. $V(x,y,z) = e^{xy}\vec{i} - e^{xy}\vec{j} + e^{yz}\vec{k}$.
c. $V(x,y,z) = xz\vec{i} + xz\vec{j} + xy\vec{k}$.
Get *code*.

A vector field F on an open subset U on $\mathbb{R}^n$ is said to *have a vector potential*, if there exists a vector field G, called a vector potential on U, such that

$$curl(G) = \nabla \times G = F.$$

Fact 8. (existence of a vector potential, [11, p.61])
$F = (f_1, ..., f_n) : U \to \mathbb{R}^n$ has a vector potential $\Leftrightarrow \nabla \bullet F = \mathrm{div}(F) = 0$.

Exercise 41. (vector potential, cf. [11, p.62])
a. Verify that $F(X) = \frac{X}{\|X\|^3}$ for $X := (x,y,z)$ has a vector potential on $\mathbb{R}^3 \setminus \mathtt{z-axis}$, namely

$$G(x,y,z) := \frac{(yz, -xz, 0)}{(x^2+y^2)(x^2+y^2+z^2)^{1/2}}$$

b. Check the existence and, if applicable, find a scalar potentail for the vector fields . $F(x,y,z) = (xy, yz, zx)$ and $G(x,y,z) = (y, \cos yz, y\cos yz)$.
Get *code*.

Exercise 42. (using REDUCE package `Avector`)
Solve exercise.45 using REDUCE package `Avector`.
Get *code*.

Exercise 43. (from cf. ARNOLD [3, p.194])
Show that **curl grad** = div **curl** =0.
Give a mathematical proof and a verification using REDUCE.

- This statement by ARNOLD is a bit misleading. Why?
  Try to interpret his proposition and then verify it using REDUCE and REDUCE$^{\texttt{excalc}}$.

11 Pullback

Q: What is a pullback?
A: Nothing but a change of variables.

Differential forms are the objects we integrate on oriented manifolds in a coordinate-independent way. Therefore we abstract the well-known definition from Calculus I

$$\int_a^b f(x)dx \quad \Rightarrow \quad \int_M \omega = ?? \tag{11.1}$$

where we now interpret the integrand $f(x)dx$ as a 1-form in $\Omega^1\mathbb{R}^1$.
In the following sections we will integrate

1. 1-forms over 1-dimensional oriented manifolds, i.e. $M = [a, b] \subset \mathbb{R}$,
2. 2-forms over 2-dimensional oriented manifolds, i.e. a parameterized surface $M \subset \mathbb{R}2$,
3. 3-forms over 3-dimensional oriented manifolds, i.e. a volume M in $\mathbb{R}^3$,
4. k-forms over k-dimensional oriented manifolds M

and we have to give the notation $\int_M \omega = ??$ an explicit 'working' meaning.
As an essential prerequisite we first discuss the concept of a *pullback*.

Mental image

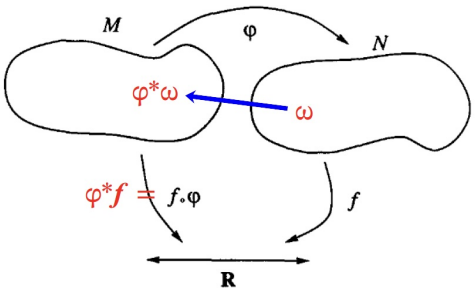


Figure 12: Preview: Pulling back function f and differential form ω from N to M. In the following sections we will successively give meaning to this diagram.
[picture enhanced from stackexchange]

11.1 Change of variables

Double and triple integrals are evaluated by 'sum'ing up the values of real-valued functions of two resp. three variables. Integrals in polar, cylindrical or spherical coordinates make thereby often use of symmetries to simplify a calculation. For example, the double integral $\iint_M f(x,y)\,dx\,dy$ may be thought of as a sum of the values of $f(x,y)$ over an infinitesimal rectangle with area $dx \cdot dy$. Going to other coordinate systems these infinitesimal regions change appropriate, i.e. in the case of polar coordinates in the plane the new region is $r\,dr\,d\theta$. This term reminds of the 2-form $r\,dr \wedge d\theta$.

Here is the general pattern of this *change of variables for a 2-dimensional integral:*

Fact 9.

$$\iint_U f(x,y)\,dx\,dy = \iint_{U^*} f_{(x(u,v),y(u,v))} \left|\frac{\partial(x,y)}{\partial(u,v)}\right| du\,dv \tag{11.2}$$

where

$$\left|\frac{\partial(x,y)}{\partial(u,v)}\right| := abs(det(\begin{bmatrix}\frac{\partial x}{\partial u} & \frac{\partial y}{\partial u}\\ \frac{\partial x}{\partial v} & \frac{\partial y}{\partial v}\end{bmatrix})) = \left|\frac{\partial x}{\partial u}\cdot\frac{\partial y}{\partial v} - \frac{\partial x}{\partial v}\cdot\frac{\partial y}{\partial u}\right| := Jac^{(x,y)}_{(u,v)} \tag{11.3}$$

is the absolute value of the so-called JACOBI *determinant* of the coordinate transformation rule $(u,v) \in U^* \mapsto (x,y) \in U \subset \mathbb{R}^2$.

Example 23. (verification of change of variables formula using REDUCE$^{\texttt{excalc}}$)
a. Verify formula (11.2) with REDUCE$^{\texttt{excalc}}$.
b. Show the change of variables formula for polar coordinates u and v.
c. Calculate $\iint_{D_a} e^{-(x^2+y^2)}\,dx\,dy$, where D_a is the disk $x^2+y^2 \le 4$.[31]

Solution **a.**

For a full understanding you may send each code line for itself to the InputDisplay:

```
load_package excalc;
     %  Declare coordinates and functions as 0-forms
pform {x,y, u,v, f}=0;
fdomain x=x(u,v), y=y(u,v);
     % calculate JACOBI determinant
d x;
d y;
d x ^ d y;
jac1 := det mat( (@ (x,u),@ (x,v)), (@ (y,u),@ (y,v) );
jac2 := @ (x,u) * @ (y,v) - @ (x,v) * @ (y,u);
```

The REDUCE output verifies (11.3):

$$du \wedge dv \left(\frac{\partial x}{\partial u}\frac{\partial y}{\partial v} - \frac{\partial x}{\partial v}\frac{\partial y}{\partial u}\right)$$

Get *code*.

[31] cf. Marsden [31, p.870]. There is no direct way of evaluate this integral using xy cordinates!

Solution **b.**

```
load_package excalc;
     % 1. New and old coordinates as 0-forms
pform {x,y}=0, {r,th}=0;
x := r*cos(th);
y := r*sin(th);
for all th let sin(th)**2+cos(th)**2=1;
     % 2. Pull back the area form
omega := d x ^ d y;
     % 3. Extract the Jacobian via interior products
jac := @th _| (@r _| omega);
```

$jac := r$

Get *code*.

Solution **c.**

The disc is given by $0 \leq r \leq a$ and $0 \leq \theta \leq 2\pi$ in polar coordinates (r, θ).
We then have: $\iint_{x^2+y^2 \leq a^2} e^{-(x^2+y^2)}\, dx\, dy = \int_0^{2\pi} \int_0^a e^{-r^2} r\, dr\, d\theta = \cdots = \pi(1 - e^{-a^2})$.
Using the area element for polar coordinates $jac := r$, we have

```
% solution provided by F.J. Wright
load_package avector;
getcsystem cylindrical;
on rounded;
surfint(e^(-r^2), avec(r,0,phi), r=0 .. 2, phi=0 .. 2pi);

off rounded;
surfint(e^(-r^2), avec(r,0,phi), r=0 .. 2, phi=0 .. 2pi);
```

3.084

Get *code*.

Remark. a step-by-step solution by pure REDUCE would be:

```
% define transformation (r,theta) -> (x,y)
x := r * cos(theta);
y := r * sin(theta);

% declare Jacobi matrix and fill it with partial diff's
matrix jacobi(2,2);
   jacobi(1,1) := df(x, r);
   jacobi(1,2) := df(x, theta);
   jacobi(2,1) := df(y, r);
```

```
   jacobi(2,2) := df(y, theta);

% calculate JACOBI determinant jac
for all x let sin(x)^2+cos(x)^2=1;
jac := det(jacobi);

% function f to be integrated
f := exp(-(x^2 + y^2));

% the new area element is f * jac
integrand := f * jac;
result := int( int( integrand, r, 0, 2), theta, 0, 2*pi);

on rounded$
print_precision 5$
int(int(exp(-r^2)*r, r,0,2), th,0,2*pi);
```

3.084

REDUCE output for (b):

```
for all th let sin(th)**2+cos(th)**2=1;

    % 2. Pull back the area form
omega := d x ^ d y;
```

$$\omega := d\,r \wedge d\,\mathrm{th}\,r$$

```
    % 3. Extract the Jacobian via interior products
jac := @th _| (@r _| omega);
```

$$\mathrm{jac} := r$$

```
on rounded$

print_precision 5$

int(int( exp(-r**2) * r, r,0,2), th, 0,2*pi);
```

$$3.0841$$

Get *code*.

11.2 The pullback

We start with three concrete examples of pullbacks to help you get used to it. A 'pullback' describes how integrals transform under changes of variables, i.e. it captures the change of variables formula for integrals.

Example 24. (pullback as 'change of variable')
We do an integration by substitution.
Task: $\int_2^3 \sin(x^2)\ 2x\,dx = ?$

Solution:

1st we solve the problem using elementary calculus.

1. $f(u) = \sin(u)$ and $g(x) = x^2$ and $g'(x) = 2x$. define functions
2. $u := x^2$ with $du = 2xdx$ substitution of inner function
3. $$\begin{aligned} \int_2^3 \sin(x^2)\ 2x\,dx &= \int_2^3 f(g(x)) \cdot g'(x)\,dx && \text{use definition} \\ &= \int_4^9 f(u)\,du && \text{transform under changes of variables} \\ &= \int_4^9 \sin(u)\,du = [-\cos(u)]_4^9 \approx 0.2575 \end{aligned}$$

2nd we solve the problem using core REDUCE following the above steps.

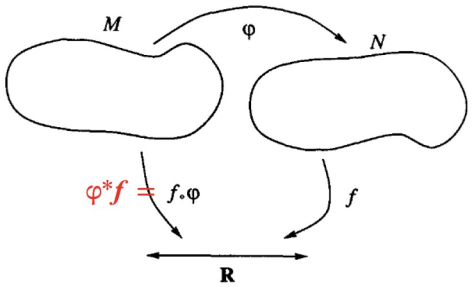
```
int(sin(x**2)*2*x, x,2,3);          %  - cos(4) + cos(9)
on  rounded$
print_precision 5$
int(sin(x**2)*2*x, x,2,3);          %  0.2575

int( sin(u), u,4,9);                %  0.2575
```

Get *code*.

11.2.1 Pullback of a function

A particular important case of a pullback is that of differential forms, e.g. if $f \in \Omega^0 N$.



Pulling back f from N to M.

We *pull back a function* $f : N \to \mathbb{R}$ by changing this function to a function $\varphi^* f : M \to \mathbb{R}$ using the composition $\varphi^* f := f \circ \varphi$ where $\varphi^* f$ is called the *pullback of* f *by* $\varphi : M \to N$.

Example 25. (pullback of a function, i.e. a 0-form, from [9, p.80])
Let $M_{(u,v)}$ and $N_{(x,y)}$ be manifolds in $\mathbb{R}^2$ with coordinates (u, v) resp. (x, y).
Let $\varphi : M_{(u,v)} \to N_{(x,y)}$ be defined by $(u, v) \mapsto (2u - v, 3u + 2v)$.
Task: Find the pullback $\varphi^* f$ of $f(x, y) := x^2 - y^2$.

Solution:

1st mathematically.

$$\varphi^* f(u, v) \overset{p.d.}{=} f \circ \varphi(u, v) = (2u - v)^2 - (3u + 2v)^2 = -5u^2 - 6uv - 3v^2$$

2nd we solve the problem using REDUCE$^{\texttt{excalc}}$.

```
load_package excalc;

% Declare coordinates and functions as 0-forms
pform {x,y, u,v} =0, f=0; phi=0;

% Define implicit the components of function phi
x := 2*u-v;
y := 3*u+2*v;

% the pullback phi*f = f o phi is automatically and implicitly done with
f := x**2 - y**2;
```

$-5u^2 -^6 uv - 3v^2$

Get *code*.

As a surplus we calculate the pullback of f explicit with the REDUCE package `EDS` and very short and esthetic with the help of file `mappings.red` by E. SCHRÜFER.

3rd we solve the problem using REDUCE^{EDS}.

```
% 1. Load Exterior Differential Systems package EDS (Harper)
load_package eds;

% 2. Declare coordinates as 0-forms
pform {x,y,z, u,v} = 0;

% 3. Declare f as 0-form
pform f = 0;
f := x**2 - y**2;

% 4. Define map phi as substitution list
phi := {x = 2*u-v, y = 3*u+2*v};

% 5. Calculate the pullback with EDS-operator pullback(Form, Map)
pullback(f, phi);
```

REDUCE output for (b):

```
f := x**2 - y**2;
```

$$f := x^2 - y^2$$

```
% 4. Define map phi as substitution list
phi := {x = 2*u-v, y = 3*u+2*v};
```

$$\phi := \{x = 2\,u - v\,,\ y = 3\,u + 2\,v\}$$

```
% 5. Calculate the pullback with EDS-operator pullback(Form, Map)
pullback(f, phi);
```

$$-5\,u^2 - 16\,u\,v - 3\,v^2$$

Get *code*.

4th we solve the problem using the REDUCE file `mappings.red`[32] by E. SCHRÜFER.

```
in "mappings.red"$   % E. Schruefer

pform {x,y,z, u,v}=0;
mapping phi: (u,v) -> (x=2*u-v,y=3*u+2*v);
phi(*,x**2-y**2);
```

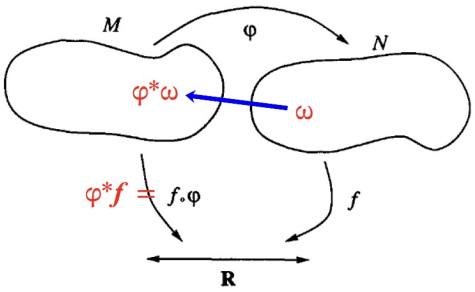
$-5u^2\ {}^{-6}\ uv - 3v^2$

Get *code*.

[32]this file was contributed by E. SCHRÜFER and is downloadable here: `mappings.red`

11.2.2 Pullback of a k-form

Now we handle pullbacks of differential k-forms, e.g. $\omega \in \Omega^k N$.



Pulling back ω from $\Omega^k N$ to $\Omega^k M$.

Let's pull back a 2-form ω by using a change of variables ϕ from (u, v) to (x, y).
We express the 2-form $\omega = f(x, y)\, dx \wedge dy$ explicitly by

$$f_{(x(u,v),y(u,v))} \left| \frac{\partial(x, y)}{\partial(u, v)} \right| du\, dv$$

using a function $\phi^* f : M \to \mathbb{R}^2$ and the composition $\phi^* f := f \circ \phi$, called the *pullback of* f *by* $\phi : \mathbb{R}^2_{(u,v)} \to \mathbb{R}^2_{(x,y)}$.

Example 26. (pullback of a 2-form,, from [9, p.81])
Let $M_{(u,v)}$ and $N_{(x,y,z)}$ be manifolds in $\mathbb{R}^2$ resp. $\mathbb{R}^3$ with coordinates (u, v) resp. (x, y, z).
Let $\phi : M_{(u,v)} \to N_{(x,y,z)}$ be defined by $(u, v) \mapsto \phi(u, v) := (u^2, uv, v^2) = (x, y, z)$.
Task: Find the pullback $\phi^*\omega \in \Omega^2\mathbb{R}^2$ of $\omega = y \cdot dx \wedge dy \in \Omega^2\mathbb{R}^3$.

Solution:

a. with the REDUCE package `EDS` .

```
load_package eds;
pform {x,y,z, u,v, f} = 0;
pform w = 2;
w := y * (d x ^ d z);
phi := {x = u**2, y = u*v, z = v**2};
pullback(w, phi);
```

Get *code*.

b. with the file `mappings.red`.

```
in "mappings.red"$
pform {u,v, x,y,z}=0;
pform w = 2;
w := y * (d x ^ d z);
mapping phi: (u,v) -> (x = u**2, y = u*v, z = v**2);
phi(*,w);
```

$4 \cdot u^2 v^2 \cdot (du \wedge dv)$

Get *code*.

Exercises

Exercise 44.
a. Verify the REDUCE calculation by a mathematical derivation.
b. Reproduce the calculation using plain REDUCE`excalc`, i.e. without operator `pullback`.

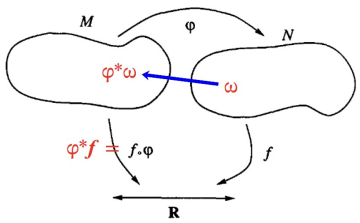
Exercise 45. (from Arnold [3, p.183, example])
If $y = f(x_1, x_2) = x_1^2 + x_2^2$ and $\omega = dy$, calculate $f^*\omega$.

$f^*\omega = 2x_1 dx_1 + 2x_2 dx_2$

11.2.3 Properties of Pullbacks

The pullback of differential forms has three properties which make it extremely useful.

Fact 10. (*properties of the pullback* ϕ^*)
Let $\phi : M \to N$ be a smooth map between smooth manifolds M and N.
Let $f : N \to \mathbb{R}$ be a smooth function on N.
Then the *pullback* ϕ^* *of* f *by* ϕ is the smooth function $\phi^* f$ on M defined by
$(\phi^* f)(x) := f(\phi(x))$. The pullback $\phi^* f$ is compatible with



(1) additivity, i.e.
For two k-forms α, β on N,

$$\phi^*(\alpha + \beta) = \phi^*\alpha + \phi^*\beta \in \Omega^k(M) \quad (11.4)$$

(2) the wedge product, i.e.
For two k-forms α, β on N,

$$\phi^*(\alpha \wedge \beta) = \phi^*\alpha \wedge \phi^*\beta \in \Omega^k(M) \quad (11.5)$$

(3) the exterior derivative, i.e.
if α is a differential k-form on N then

$$\phi^*(d\alpha) = d(\phi^*\alpha) \in \Omega^k(M) \quad (11.6)$$

- FYI: pullback

11.3 Definition of the Integral $\int_M \omega$

> The integral of a differential form is a higher-dimensional generalization of such ideas as the flux of a fluid across a surface or the work of a force along a path. cf. [3, p.181]

We integrate differential forms on oriented manifolds in a coordinate-independent way. Therefore we define

Definition (*Integral of a k-form*)**.**
Let M be a k-dimensional manifold in $\mathbb{R}^n$ with $k < n$.
Let $\phi : U \subset \mathbb{R}^k \to M$ be a smooth parameterization with coordinates $u^1, ..., u^k$ of M.
Let $\omega \in \Omega^k M$ be a differentiable k-form on M.
Then

$$\int_M \omega := \int_U \phi^* \omega := \int_U \omega(\frac{\partial \phi}{\partial u^1}, \dots, \frac{\partial \phi}{\partial u^k}) du^1 \dots du^k \tag{11.7}$$

where $\omega(\frac{\partial \phi}{\partial u^1}, \dots, \frac{\partial \phi}{\partial u^k})$ is the action of the k-form ω on the tangent vectors $(\phi_{u^1}, \dots, \phi_{u^k})$, see §7 and §8.

The following examples should make this construction comprehensible.

Example 27. (from Bryan [8, p.5])
Let M be the 1-manifold in $\mathbb{R}^3$ defined by the parameterization $\phi(t) = (3t, t^2, 5 - t)$ for $0 \le t \le 2$. – You can see that M is just a curve C in three dimensions.
Let ω be the 1-form $\omega = 2y \cdot dx - xz \cdot dy + dz$.
Find $\int_C \omega$.

Solution

1st mathematically.

$$\begin{aligned}
\int_M \omega &= \int_{[0,2]} \phi^* \omega \\
&= \int_0^2 \omega(\frac{\partial \phi}{\partial t}) dt \overset{1}{=} \int_0^2 \omega(3, 2t, -1)\ dt \\
&\overset{2}{=} \int_0^2 (2y \cdot 3 - xz \cdot 2t - 1)\ dt \\
&\overset{3}{=} \int_0^2 (6t^2 - (30t^2 - 6t^3) - 1)\ dt = \dots = -42
\end{aligned}$$

1 : because ω acts on tangent vector ϕ_t

2 : evaluate ω on $\phi_t = (3, 2t, -1)$, i.e. $\omega \lrcorner (3, 2t, -1)$

3 : substitute $y = t^2, x \cdot z = (3t) \cdot (5 - t) = 15t - 3t^2$ in 2.

2nd using REDUCE package EDS.

```
% 1. calculate integrand, i.e. pullback omega_*_phi
load_package eds;
pform {x,y,z, t_} = 0;
pform w = 1;
w := 2*y* d x - x*z* d y + d z;
phi := {x = 3*t_, y = t_**2, z = 5-t_};
pullback(w, phi);

% 2. integrate omega_*_phi
int(6*s^3-24*s^2-1, s,0,2);
```

-42

Get *code*.

Example 28. (from BRYAN [8, p.5])
Let M be a 2-manifold in $\mathbb{R}^4$ with coordinates[33] x^1, x^2, x^3, x^4 parameterized by
$\phi(u,v) = (u, u-v, 3-u+uv, -3v)$, where where $u^2+v^2 < 1$.
Take as 2-form $\omega = x^2 \cdot dx^1 \wedge dx^3 - x^4 \cdot dx^3 \wedge dx^4$.
Find $\int_M \omega$.

Solution:

1st mathematically.

$$\begin{aligned}
\int_M \omega &= \int_D \phi^*\omega = \int_D \omega(\frac{\partial\phi}{\partial u}, \frac{\partial\phi}{\partial v})du\,dv \\
&\overset{1}{=} \int_D x^2 \cdot dx^1 \wedge dx^3(\frac{\partial\phi}{\partial u}, \frac{\partial\phi}{\partial v})du\,dv - \int_D x^4 \cdot dx^3 \wedge dx^4(\frac{\partial\phi}{\partial u}, \frac{\partial\phi}{\partial v})du\,dv \\
&\overset{2}{=} \int_D x^2 \cdot \begin{vmatrix} \frac{\partial x^1}{\partial u} & \frac{\partial x^1}{\partial v} \\ \frac{\partial x^3}{\partial u} & \frac{\partial x^3}{\partial v} \end{vmatrix} du\,dv - \int_D x^4 \cdot \begin{vmatrix} \frac{\partial x^3}{\partial u} & \frac{\partial x^3}{\partial v} \\ \frac{\partial x^4}{\partial u} & \frac{\partial x^4}{\partial v} \end{vmatrix} du\,dv \\
&\overset{3}{=} \int_D x^2 \cdot \begin{vmatrix} 1 & 0 \\ v-1 & u \end{vmatrix} du\,dv - \int_D x^4 \cdot \begin{vmatrix} v-1 & u \\ 0 & -3 \end{vmatrix} du\,dv \\
&= \int_D (x^2 \cdot u - x^4 \cdot (3-3v))du\,dv \overset{4}{=} \int_D ((u-v)\cdot u - 3v \cdot (3-3v))du\,dv \\
&= \int_D (u^2 - uv + 9v - 9v^2)du\,dv = \ldots
\end{aligned}$$

1 : definition of ω and linearity of $\int$ and $\wedge$

2 : see action of 2-forms on tangent vectors in §7

3 : the tangent vectors are $\frac{\partial\phi}{\partial u} = (\frac{\partial x^1}{\partial u}, \frac{\partial x^2}{\partial u}, \frac{\partial x^3}{\partial u}, \frac{\partial x^4}{\partial u}) = (1, 1, v-1, 0)$

... and $\frac{\partial\phi}{\partial v} = (\frac{\partial x^1}{\partial v}, \frac{\partial x^2}{\partial v}, \frac{\partial x^3}{\partial v}, \frac{\partial x^4}{\partial v}) = (0, -1, u, -3)$

4 : remember: $x^2 \equiv x_2 = u - v$ is the 2nd component of ϕ.

[33] please note: the indices are written superscripted, e.g. x^2 means x_2 and not the square of x.

Now $D : x^2 + v^2 \leq 1$ is the unit circle in the uv-plane and we get

$$.. = \int_D (u^2 - uv + 9v - 9v^2) du\, dv = \int_{-1}^{1} \int_{-\sqrt{1-v^2}}^{\sqrt{1-v^2}} (u^2 - uv + 9v - 9v^2) du\, dv = -2\pi$$

2nd using REDUCE package `EDS`.
Please look, how easy it is to get the right integrand $\phi^*\omega$ using a pullback ;)

```
load_package eds;
spacedim 4;
     % 1. Declare coordinates as 0-forms
pform {x1,x2,x3,x4, u,v} = 0;
     % 2. Declare omega w as 2-form
pform w = 2;
w := x2 * d x1 ^ d x3 - x4 * d x3 ^ d x4;
     % 3. Define map phi as substitution list
phi := {x1 = u, x2 = u-v, x3 = 3-u+u*v, x4 = -3*v};
     % 4. Calculate the pullback i.e. the integrand w_*_phi
pullback(w, phi);
     % 5a. integrate w_*_phi in steps (hint by fjW)
int (u^2-u*v+9v-9v^2, u, -sqrt(1-v^2), sqrt(1-v^2));
int(ws, v, -1, 1);
     % 5b. integrate w_*_phi using polar coords and a bit trigonometry
int( int( (r**2*cos(th)**2 - 9*r**2*sin(th)**2)*r, r,0,1), th, 0,2*pi);
```

REDUCE output:

File Edit Calculus Algebra Matrix Graphs Load Package Switch

phi := {x1 = u, x2 = u-v, x3 = 3-u+u*v, x4 = -3*v};

$$\phi := \left\{x_1 = u\,,\ x_2 = u - v\,,\ x_3 = u\,v - u + 3\,,\ x_4 = -3\,v\right\}$$

pullback(w, phi);

$$d\,u \wedge d\,v\left(u^2 - u\,v - 9\,v^2 + 9\,v\right)$$

int (u^2-u*v+9v-9v^2, u, -sqrt(1-v^2), sqrt(1-v^2));

$$-18\sqrt{-2\,v+1}\,v$$

int(ws, v, -1, 1);

$$\frac{12\left(3\sqrt{3} - 2\,i\right)}{5}$$

int(int((r**2*cos(th)**2 - 9*r**2*sin(th)**2)*r, r,0,1), th, 0,2*pi);

$$-2\,\pi$$

Get *code*.

Example 29. (from BRYAN [8, p.5])
Let M be a 3-manifold in $\mathbb{R}^4$ with coordinates x_1, x_2, x_3, x_4 parameterized by
$\phi(u, v, w) = (vw, u^2, 1 - 3v + w, uv)$, where where $u^2 + v^2 + w^2 < 1$.
Let $\omega = x_3 \cdot dx_1 \wedge dx_2 \wedge dx_4 \in \Omega^3\mathbb{R}^4$ as a 3-form.
Find $\int_M \omega$.

Solution:

1st mathematically.

We use the precision of the 'action' of ω on a triple of tangent vectors[34] (ϕ_u, ϕ_v, ϕ_w) as described in §8. $D : x^2 + v^2 + w^2 \leq 1$ is the unit ball in the uvw-space.
We have

$$\begin{aligned}
\int_M \omega &= \int_D \phi^*\omega = \int_D \omega(\phi_u, \phi_v, \phi_w) du\, dv\, dw \\
&\overset{1}{=} \int_D x_3 \cdot dx_1 \wedge dx_2 \wedge dx_4(\phi_u, \phi_v, \phi_w) du\, dv\, dw \\
&\overset{2}{=} \int_D x_3 \cdot \begin{vmatrix} \partial_u x_1 & \partial_v x_1 & \partial_w x_1 \\ \partial_u x_2 & \partial_v x_2 & \partial_w x_2 \\ \partial_u x_4 & \partial_v x_4 & \partial_w x_4 \end{vmatrix} du\, dv\, dw \overset{3}{=} \int_D x_3 \cdot \begin{vmatrix} 0 & w & v \\ 2u & 0 & 0 \\ v & u & 0 \end{vmatrix} du\, dv\, dw \\
&\overset{(a)}{=} \int_D (2u^2 v - 6u^2 v^2 + 2u^2 vw) du\, dv\, dw \\
&\overset{4}{=} \int_0^{\pi} \int_0^{2\pi} \int_0^1 (-6r^6 \sin(\phi)^5 \cos(\theta)^2 \sin(\theta)^2)\, dr\, d\theta\, d\phi = -\frac{8}{35}
\end{aligned}$$

1 : definition of ω

2 : see action of 3-forms on tangent vectors in §8

3 : the tangent vectors are $\partial_u\phi = (\partial_u x_1, \partial_u x_2, \partial_u x_3, \partial_u x_4) = (0, 2u, 0, v)$

... and $\partial_v\phi = (w, 0, -3, u)$, $\partial_w\phi = (v, 0, 1, 0)$

4 : switching to spherical polar coordinates $u = r\sin\phi\cos\theta$, $v = r\sin\phi\sin\theta$ and $w = r\cos\theta$

... with volume element $du\, dv\, dw = r^2 \sin\phi\, dr\, d\theta\, d\phi$ and doing a bit trigonometry

2nd using REDUCE package `EDS`.

It is easy to get the right integrand $\phi^*\omega$ using a pullback, because it automates the first three steps. Calculate the triple integral in line 4 of the derivation above with spherical coordinates.

```
load_package eds;
spacedim 4;
pform {x1,x2,x3,x4, u,v,w} = 0;

pform omega = 3;
```

[34] remember the other notations: $\phi_u \equiv \frac{\partial\phi}{\partial u} = \partial_u\phi$.

```
omega := x3 * d x1 ^ d x2 ^ d x4;
phi := {x1 = v*w, x2 = u**2, x3 = 1-3*v+w, x4 = u*v};

% calculate the integrand as pullback
pullback(omega, phi);

% integrate w_*_phi using the polar coords version in 4. in stages
int( -(6*r**6)*sin(ph)**5*cos(th)**2*sin(th)**2,  r,0,1);
int(ws,  th, 0, 2*pi);
int(ws,  ph, 0,   pi);
```

REDUCE output:

```
phi := {x1 = v*w, x2 = u**2, x3 = 1-3*v+w, x4 = u*v};
```

$$\phi := \left\{x_1 = v\,w\,,\ x_2 = u^2\,,\ x_3 = -3\,v + w + 1\,,\ x_4 = u\,v\right\}$$

```
% calculate the integrand as pullback
pullback(omega, phi);
```

$$2\ d\,u \wedge d\,v \wedge d\,w\,u^2\,v\,(-3\,v + w + 1)$$

```
int( -(6*r**6)*sin(ph)**5*cos(th)**2*sin(th)**2,  r,0,1);
```

$$\frac{-6\cos(\mathrm{th})^2\sin(\mathrm{ph})^5\sin(\mathrm{th})^2}{7}$$

```
int(ws,  th, 0, 2*pi);
```

$$\frac{-3\sin(\mathrm{ph})^5\pi}{14}$$

```
int(ws,  ph, 0,   pi);
```

$$\frac{-8\,\pi}{35}$$

Get *code*.

Exercises

Exercise 46. Integrate the triple integral in version (a) using appropriate integration limits.

Exercise 47. (from [3, p.187])
Let M be the plane $\mathbb{R}^2_{(p,q)}$ and ω_1 the form $p\,dq$ and C the unit circle in $\mathbb{R}^2$.
Let $p = \cos t, q = \sin t$ where $t \in [0, 2\pi]$. Verify: $\int_C p\,dq = \pi =$ area of the unit circle..

Remark: In general, if C represents the boundary of a region D, then the integral of the form $p\,dq$ over the boundary $\int_C p\,dq$ of the region is equal to the area of region D.

11.4 Exercises



We do some selected examples and exercises from [4] with REDUCE^excalc.
The reader may consult the book by BACHMAN for the mathematical considerations.

Exercise 48. (from [4, p. 44])
Let $\omega = xy^2 dx \wedge dy$ be a differential 2-form on $\mathbb{R}^2$. Let $D \subset \mathbb{R}^2$ be the region $[0,1] \times [0,1]$.
Calculate $\int_D \omega$. $\frac{1}{6}$

Exercise 49. (from [4, p. 48])
Let M denote the top half of the unit shpere in $\mathbb{R}^3$.
Let $\omega = x^2 dx \wedge dy$ be a differential 2-form on $\mathbb{R}^3$. Calculate $\int_M \omega$. $\frac{\pi}{2}$

Hint: use the parameterization $\phi(r,t) = (r\cos t, r\sin t, \sqrt{1-r^2})$ where $0 \le t \le 2\pi$ and $0 \le r \le 1$.

Exercise 50. (from [4, p. 48]
Integrate the 2-form $\omega = \frac{1}{x} dy \wedge dz - \frac{1}{y} dx \wedge dz$ over the surface parameterized by $\phi(r,\theta) = (r\cos\theta, r\sin\theta, \cos r)$, $0 \le r \le \pi/2$ and $0 \le \theta \le 2\pi$. 4π

Exercise 51. (from [4, p. 49])
Let S be the surface parameterized by $\phi(u,v) = (2u, v, u^2+v^3)$, $0 \le u \le 1$ and $0 \le v \le 2$.
Integrate the 2-form $(x+2y)dx \wedge dz$ over S. 64

Exercise 52. (from [4, p. 52])
Let S be the frustrum parameterized by $\phi(r,\theta) = (r\cos\theta, r\sin\theta, r))$, $0 \le r \le 2$ and $0 \le \theta \le 2\pi$. Integrate the 2-form $\omega = z dx \wedge dy + \frac{1}{x} dy \wedge dz + \frac{1}{y} dx \wedge dz$ over S with the orientation induced by θ. $\frac{14\pi}{3}$

Exercise 53. (from [4, p. 54]
Let C be the curve in $\mathbb{R}^2$ parameterized by $\phi(t) = (t^2, t^3)$ where $0 \le t \le 2$. Let $\omega = ydx + xdy$ be a given 1-form. Calculate $\int_C \omega$. 32

Exercise 54. (from [4, p. 54]) Let C be the curve in $\mathbb{R}^3$ parameterized by $\phi(t) = (2\cos t, 2\sin t, t^2)$ where $0 \le t \le 2$. Integrate the 1-form $(x^2+y^2)dz$ over C. 16

Exercise 55. (from [4, p. 56])
Let $Q := \{(x,y,z) | 0 \le x,y,z \le 1\}$ be the unit cube in $\mathbb{R}^3$.
Integrate the 3-form $x^2 z dx \wedge dy \wedge dz$ over Q. $\frac{1}{6}$

Exercise 56. (from [4, p. 56])
Suppose $\phi(a,b,c) := (a+b, a+c, bc, a2)$, where $0 \le a,b,c \le 1$.
Let M be the image of ϕ with the induced orientation.
Suppose $\omega = dy \wedge dz \wedge dw - dx \wedge dz \wedge dw - 2ydx \wedge dy \wedge dz$. Calculate $\int_M \omega$. $\frac{7}{6}$

Exercise 57. (from [4, p. 58])
We will use the parameterization $\phi(u,v) = (u, u^2 + v2)$ to evaluate the double integral $\iint_R (x^2+y)dA$ over the region R of the xy-plane bounded by the parabolas $y = x^2$ aand $y = x^2 + 4$ and the lines $x = 0$ and $x = 1$. $\frac{32}{3}$
Draw a picture of the situation.

Exercise 58. (from [4, p. 58])
Let R be the region of the xy-plane bounded by the ellipse $9x^2 + 4y^2 = 36$.
Calculate $\int_R x^2 dx \wedge dy$ $\frac{32}{3}$
Draw a picture of the situation.

Exercise 59. (from [4, p. 58])
Use the parameterization $\phi(u, v) = (u, u^2+v2)$ to evaluate the double integral $\iint_R (x^2+y)dA$ over the region R of the xy-plane bounded by the parabolas $y = x^2$ and $y = x^2 + 4$ and the lines $x = 0$ and $x = 1$. $\frac{32}{3}$
Draw a picture of the situation.

Exercise 60. (from [4, p. 63])
Let V be the region between the cylinders of radii 1 and 2 and between the planes $z = 0$ and $z = 1$. i.e. $V = \{(r, \Theta, z) | 1 \leq r \leq 2, 0 \leq z \leq 1\}$.
Calculate $\int_V z(x^2 + y^2)\ dx \wedge dy \wedge dz$ $\frac{15\pi}{4}$

Remark: The region V is best parameterized using cylindrical coordinates:
$\phi(r, \Theta, z) = (r \cos \Theta, r \sin \Theta, z)$, where $1 \leq r \leq 2, 1 \leq \Theta \leq 2\pi, 0 \leq z \leq 1$.

12 Integration on Manifolds aka Vector Analysis

In core REDUCE `int(f,x)` returns the indefinite integral $\int f$ of f with respect to x and `int(f, x,a,b)` the definite integral $\int_a^b f$ with limits a and b. To begin with, we present some examples for the use of the integration operator `int()` of REDUCE.

12.1 Integration with REDUCE

Example 30. Calculate the indefinite integral $\int x^2\,dx$ and the definite integral $\int_1^2 x^2\,dx$.

Solution:

```
int(x^2,x);
int(x^2,x,1,2);
```

$\frac{1}{3}x^3$

Get *code*.

Example 31. (Extend the argument list for multiple integrals)
Example: Calculate the indefinite integral $\iint xy\,dx\,dy$ and the definite integral $\int_0^1 \int_2^3 xy\,dx\,dy$.

Solution 1:

```
f := x*y;
int(int(f,x),y);

f := x*y;
int(int(f,x,0,1),y,2,3);
```

$\frac{1}{4}x^2y^2$

Get *code*.

Alternatively we may use the `defint(f,x,a,b)` operator of package `Avector`, which computes the definite integral of f with respect to x evaluated from a to b.
The argument list can be stacked for multiple integrals.

Solution 2:

```
load_package avector;
f := x*y;
defint (defint(f,x,0,1), y,2,3);
```

$\frac{5}{4}$

Get *code*.

Example 32. The following example computes the integral of $f = x^2$ over the domain of a semicircle. For each x along the abscissa, y ranges from 0 to $\sqrt{1-x^2}$.

Solution 1:

```
int(int(x^2, x,-1,1), y,0,sqrt(1-x^2));    % incorrect order of variables
int(int(x^2, y,0,sqrt(1-x^2)),  x,-1,1);   % correct order
```

$\frac{1}{8}\pi$

Get *code*.

Solution 2: Alternatively, `sub` can be used to compute a definite integral step by step.

```
I1 := int(x^2,y);                                % 1st indefinite  integral
I1 := sub(y=sqrt(1 - x^2), I1) - sub(y=0, I1);
I2 := int(I1,x);                                 % 2nd  indefinite  integral
sub(x=1,I2) - sub(x=-1, I2);
```

Get *code*.

Example 33. Here is a useful trick.
Integrals involving sine and cosine can often be solved using exponentials.
We use the fact: $\sin(x) = \frac{\exp(ix)-\exp(-ix)}{2i}$ and $\cos(x) = \frac{\exp(ix)+\exp(-ix)}{2}$.
Then, for example, the definite integral

$$\int_0^{2\pi} \left(\sin^4 t - 2\cos^3(t/2)\sin t\right) dt$$

can then be solved as follows.[35]

Solution:

```
% 1. using exp-form of term
for all x let sin(x) = (exp(I*x) - exp(-I*x))/(2*I);
for all x let cos(x) = (exp(I*x) + exp(-I*x))/2;

f := sin(t)^4 - 2 cos(t/2)^3 sin(t);
int(f, t, 0, 2 *pi);

% 2. alternatively using core REDUCE
int(sin(x)^4 -2*cos(x/2)^3*sin(x), x,0,2*pi);
```

$\frac{3}{4}\pi - \frac{16}{5}$

Get *code*.

[35] F.J. WRIGHT pointed out that REDUCE does this automatically. So you may cancel the first 3 lines.

12.2 Arc length

Fact 11. [36] Let $g(t)$ be a parametric function that draws a curve in $\mathbb{R}^n$.
The arc length from $g(a)$ to $g(b)$ is given by

$$\int_a^b |g'(t)|\, dt \tag{12.1}$$

where $|g'(t)|$ is the length of the tangent vector of $g(t)$.[37]

Mental image

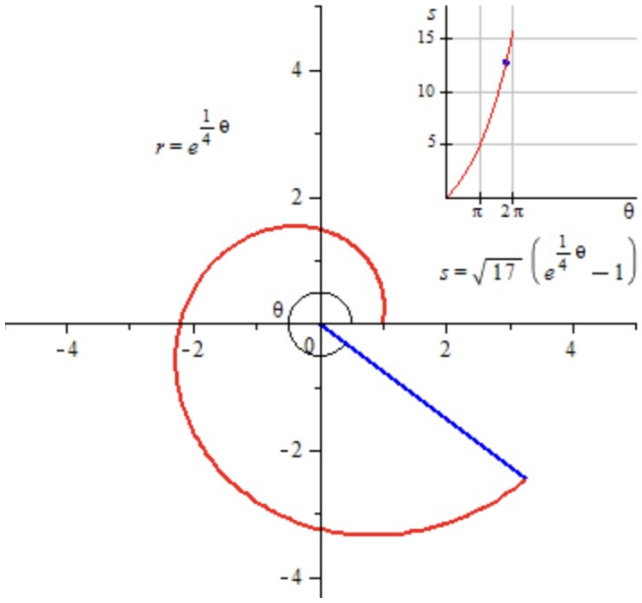


Figure 13: The arc length of the spiral $r = \exp(\frac{1}{4}\theta)$ for $\theta = 2\pi$ is approx. 15.
[from WikiPediA arc length with an animated version of this static picture.]

Example 34. Find the length of the curve $y = x^2$ from $x = 0$ to $x = 1$.

[36] The following sections §12.2– §13 with its introducing facts are taken from the EIGENMATH manual [49], the examples are transferred to REDUCE and exercises are added. Also the new perspective of differential forms is added.

[37] For details see WikiPediA: ArcLength

Solution:

1st with classic vector analysis.

```
% Load the vector calculus package
load_package avector;

% Define the curve as a 3D vector parameterized by x
g := avec(x, x^2, 0);

% Differentiate the curve w.r.t. x to get the tangent vector
dg := df(g, x);

% Compute the scalar magnitude (vector modulus) of the tangent vector
define mag = vmod;
ds := mag dg;

% Evaluate the definite integral from x = 0 to x = 1
arc_length := defint(ds, x, 0, 1);

on rounded;
print_precision 5;
arc_length;
```

$\frac{1}{2}\,5^{1/2} - \frac{1}{4}\log(2) + \frac{1}{4}\log(2\,5^{1/2} + 4)$
1.47894

Get *code*.

As expected, the result is greater than $\sqrt{2} \approx 1.414$, the length of a straight line from $(0, 0)$ to $(1, 1)$.

2nd using package `avector` of REDUCE, but with function `deflineint()`.

```
load_package avector;
getcsystem 'cartesian;

% Define the parametric path g of the curve
g := avec(x, x^2, 0);

% Define the vector function to yield the arc length integrand
v := avec( sqrt(1 + 4*x^2), 0, 0);

% Calculate the definite line integral from x = 0 to x = 1
deflineint(v, g, x, 0, 1);
```

Get *code*.

3rd *Solution* using differential forms with REDUCE$^{\texttt{excalc}}$.

The usual formula for the arc length L of a function $y = f(x)$ is: $L = \int_a^b \sqrt{1 + \left(\frac{dy}{dx}\right)^2}\, dx$. For our function $y = f(x) = x^2$ using $\frac{dy}{dx} = 2x$, we get $L = \int_0^1 \sqrt{1 + 4x^2}\, dx$, see [38].
Using REDUCE: When $y := x^2$ is declared, evaluating dy automatically yields the 1-form $2x\, dx$. Since division of differential forms is not defined[39], we have to use the interior product operator $\lrcorner$. Contracting the coordinate vector field $@x$ with the 1-form dy extracts the scalar coefficient: $\partial_x \lrcorner dy = 2x$ The above metric component is then reconstructed as $\sqrt{1 + (\partial_x \lrcorner dy)^2}$ and passed to the standard integration operator.

```
load_package excalc;

% Define coordinates as 0-forms
pform x=0, y=0;

% Define the curve term
y := x**2;

% Calculate dy/dx using _| of vector field @x and 1-form d y
dydx := @ x _| d y;

% Calculate the scalar line element ds
ds := sqrt(1 + dydx**2);

% Integrate to find the exact arc length
arc_length := int(ds x, 0, 1);

% Use rounded mode for numerical approximation
on rounded;
arc_length_approx := int(ds, x, 0, 1);
```

Get *code*.

Example 35. The following script does a discrete computation of the arc length by dividing the curve into 100 pieces.

Solution 1

```
% Number of pieces n and step size h
n := 100;
dx := 1.0/n;

% Initialise sum
length := 0.0;
```

[38] e.g. ArcLength, [31, p. 745 ff]
[39] but sometimes nevertheless possible, see examples below

```
% Loop over the 100 segments
for i := 0 : n-1 do
    <<  x1 := i*dx;      y1 := x1^2;
        x2 := (i+1)*dx;  y2 := x2^2;
        length := length + sqrt( (x2-x1)^2 + (y2-y1)^2 );
      >>;
length;

on rounded;
print_precision 5$
length;
```

1.47894 – As expected, the discrete result matches the analytic result.

Get *code*.

Solution 2 (alternatively using `hypot(x,y)`)
The hint to the following variant of a more accurate and robust solution is by F.J. Wright.

```
on rounded;
print_precision 5;
n := 100;  h := 1.0/n; length := 0.0;
for i := 0 : n-1 sum
     << x1 := i*h; x2 := (i+1)*h; hypot(x2-x1, x2^2-x1^2) >>;
```

1.47894

Get *code*.

Solution 3 (alternatively using package `listvecops`)

Remark. For the following variant we use the package `listvecops` to define the parameterization `g(t)={t,t^2}` as a pair list. In doing so, we have to use the declaration `listproc` instead of the usual declaration `procedure`. We also have to define our listvecops-version of the absolute function `abs()`, which is done using the listvecops-version of the inner product operator `*.` , i.e. `labs(v)`$= \sqrt{v_1^2 + v_2^2 + \ldots}$. So we get

```
load_package listvecops;

listproc labs v; sqrt(v *. v);   % list version of abs
listproc g x; {x,x^2};

on rounded;
print_precision 5$
procedure h k; labs (g(k/100.0)-g((k-1)/100.0));
for k:=1:100 sum h(k);
```

1.47894

Get *code*.

12.2.1 Exercises



We do some selected examples and exercises from [31] with REDUCE[excalc].
The reader may consult the book by MARSDEN for the mathematical considerations.

Exercise 61.
Find the length of the curve $y = x^{3/2}$ from the origin to $x = \frac{4}{3}$.
For a solution you may think about the pseudocode

```
g := (t,t^(3/2))
defint(abs(d(g,t)),t,0,4/3)
```

$\frac{56}{27}$

Exercise 62. Find the length of the curve from fig.13 from $\theta = 0 \ldots 2\pi$.
Get the nice *code*, contributed by F.J.WRIGHT.

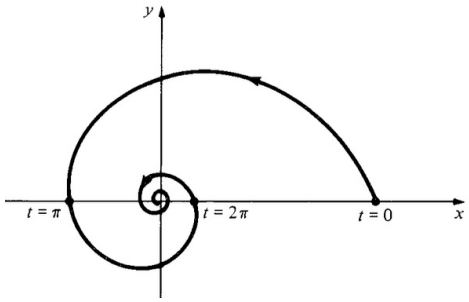
Exercise 63 (from [11], p.53)**.**
Evaluate $\int_\Gamma xydx + xz^2dy + xyzdz$ where the curve Γ is parametrized by $\phi = (t, t^2, t^3)$ where $0 \leq t \leq 1$.

Exercise 64.
Find the length of the helix $(\cos t, \sin t, t)$ for $0 \leq t \leq \pi$.

Exercise 65 (from [31], p. 746)**.**
Find the arc length of the curve $(\cos t, \sin t, t^2)$ for $0 \leq t \leq \pi$.

Exercise 66 (from [31], p. 745)**.**
Find the arc length of (e^t, t, e^t) for $0 \leq t \leq 1$.

Exercise 67 (from [31], p. 751)**.**
Find the arc length of the exponential spiral $(e^{-t}\cos t, e^{-t}\sin t, 0)$ for $0 \leq t \leq 5\pi$.



Exercise 68 (from [31], p. 745)**.**
Find the arc length of the ellipse $\frac{x^2}{a^2} + \frac{x^2}{a^2} = 1$ for $0 \leq t \leq 2\pi$.
Use the parametrization $x = a\cos t,\ y = b\sin t$.

Exercise 69 (from [31], p. 745)**.**
Find the arc length of the astroid $(x, y) = (a\cos^3 t, a\sin^3 t)$ for $0 \leq t \leq 2\pi$.

12.3 Line integral

Fact 12. (Theoretical minimum) There are two kinds of line integrals, one for scalar fields and one for vector fields. The following table shows how both are based on the calculation of arc length.

	Abstract form	Computable form
Arc length	$\int_C ds$	$\int_a^b \lvert g'(t)\rvert\, dt$
Line integral, scalar field	$\int_C f\, ds$	$\int_a^b f(g(t))\, \lvert g'(t)\rvert\, dt$
Line integral, vector field	$\int_C (F \cdot u)\, ds$	$\int_a^b F(g(t)) \cdot g'(t)\, dt$

- Note that for the measure ds we have $ds = |g'(t)|\, dt$.
- For vector fields, symbol u is the unit tangent vector $u = \frac{g'(t)}{|g'(t)|}$.
- Note that u cancels with ds as follows.

$$\int_C (F \cdot u)\, ds = \int_a^b \left(F(g(t)) \cdot \frac{g'(t)}{|g'(t)|} \right) |g'(t)|\, dt = \int_a^b F(g(t)) \cdot g'(t)\, dt \tag{12.2}$$

Mental image

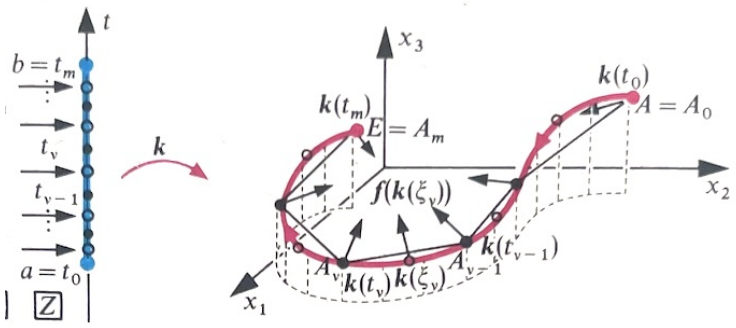


Figure 14: A vector field $\mathbf{f}$ along a curve $\mathbf{k}$ with work integral from A to E $W_{AE} = \int_a^b \mathbf{f}(\mathbf{k}(t)) \bullet \mathbf{k}'(t) dt$ [picture from [38, p.350]]

Example 36. (Line integral, scalar field).
Evaluate $\int_C x\,ds$ where C is a straight line from $(0,0)$ to $(1,1)$.

1st *Solution* with classic line integral.

```
load_package avector;
g := avec(x, x, 0);
dg := df(g, x);
define mag = vmod;
ds := mag dg;
arc_length := defint(x * ds, x, 0, 1);

on rounded;
print_precision 5;
arc_length;
```

Get *code*.

2nd *Solution* using differential forms.

```
load_package excalc;
pform {x,y,t} = 0, {dx,dy,ds} = 1;
x := t;
y := t;

@(t) _| d x;

% Line element
ds := sqrt( (@(t) _| d x)**2 + (@(t) _| d y)**2 );

% Integral
int( x*ds, t, 0, 1);
```

REDUCE output:

```
@(t) _| d x;
```

$$1$$

```
% Line element
ds := sqrt( (@(t) _| d x)**2 + (@(t) _| d y)**2 );
```

$$\mathrm{ds} := \sqrt{2}$$

```
% Integral
int( x*ds, t, 0, 1);
```

$$\frac{\sqrt{2}}{2}$$

Get *code*.

Example 37. (Line integral, vector field).
Evaluate $\int_C x\,dx$ where C is a straight line from $(0,0)$ to $(1,1)$.

Solution:

1stwith a classic line integral. We have $x\,dx = (F \cdot u)\,ds$ hence

```
load_package avector;
g := avec(x, x, 0);
F := avec(x, 0 ,0);
Line_integral := defint( F dot df(g, x), x, 0, 1);
```

$\frac{1}{2}$

Get *code*.

2nd with differential forms.

```
load_package excalc;

pform {x,y,t} = 0, {dx,dy,ds} = 1;
x := t;
y := t;

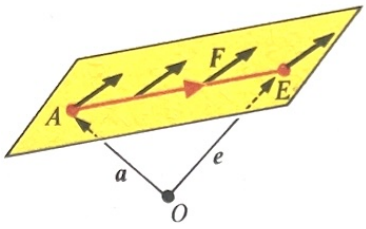
@(t) _| d x;

% Line element
ds := sqrt( (@(t) _| d x)**2 + (@(t) _| d y)**2 );

% Integral
int( x*ds, t, 0, 1);
```

$\frac{1}{2}$

Get *code*.



The work in a constant vector field $\mathbf{F}$ along a straight line from A to B is $W_{AB} = \mathbf{F} \bullet (\mathbf{e} - \mathbf{a})$.
[picture from [38, p.350]]

Example 38.
Evaluate $\int z\,dx + x\,dy + y\,dz$ along the path $x = 2t + 1$, $y = t^2$, $z = 1 + t^3$, $0 \le t \le 1$.

Solution:

1st with classic method using the given parameterization with $g = (x, y, z), F = (z, x, y)$.

```
load_package avector;
g := avec(2*s+1, s^2, 1+s^3);
F := avec(1+s^3, 2*s+1, s^2);
defint(F dot df(g,s), s, 0, 1);
```

$\frac{163}{30}$

Get *code*.

2nd differential form method.

```
% Load the EXCALC package
load_package excalc;

% Declare 0-forms (scalars/coordinates)
pform x=0, y=0, z=0, t=0, w=1;

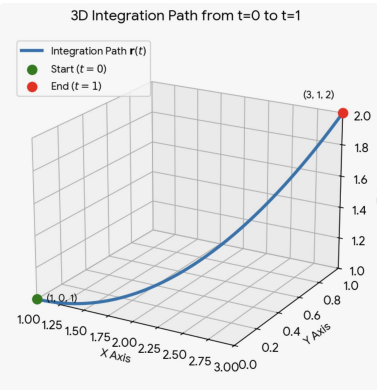
% Define the original differential 1-form omega named w
w := z*d(x) + x*d(y) + y*d(z);

% Parametrize the coordinates in terms of t
x := 2*t + 1;
y := t**2;
z := 1 + t**3;

% Evaluate w on the parameterized space (pullback to t)
w;

% Compute the line integral over t from 0 to 1
result := int(w / d(t), t, 0, 1);
```

$\frac{163}{30}$



Get *code*.

Example 39. Find the value of the integral $\int_C \mathbf{F} \bullet d\vec{r}$, where C is the semicircle parameterized by $\vec{r}(t) = (\cos t, \sin t)$, $0 \leq t \leq \pi$ and the vector field is $\mathbf{F} = (-y, x)$.

1st *Solution* with REDUCE^{avector} using classic vector analysis.

Substituting $x = \cos t$ and $y = \sin t$ into the vector field $\mathbf{F} = (-y, x, 0)$ yields $\mathbf{F}(\vec{r}(t)) = (-\sin t, \cos t, 0)$ and differentiating $\vec{r}(t)$ with respect to t gives $\vec{r}\,'(t) = (-\sin t, \cos t, 0)$:

```
% Load the vector calculus package
load_package avector;

% Define 3D vector field F in Cartesian coordinates
F := avec(-y, x, 0);

% Define the parameterized curve C using parameter t
C := avec(cos(t), sin(t), 0);

% Evaluate the definite line integral from 0 to pi
deflineint(F, C, t, 0, pi);
```

π

Get *code*.

2nd *Solution* using differential forms and REDUCE^{excalc}

```
load_package excalc;

pform x=0, y=0, t=0, omega=1;
fdomain x=x(t), y=y(t);

x := cos t;
y := sin t;

% the 1-form associated with F
omega := -y*d x + x*d y;

% automatically pull-back to the t parameter interval
omega;

% integrate the extracted coefficient of dt
%           dividing by dt is a bit abusive ;) but possible
Int( omega/d t, t, 0, pi);
% this is more 'correct'
int( coeffn(omega, d t, 1), t, 0, pi);
```

π

Get *code*.

12.3.1 Exercises

Exercise 70.
Evaluate $\int y^2\,dx$ along the straight line from $(0,0)$ to $(2,2)$.[40]
Task: give the differential form way of the following classic solution.

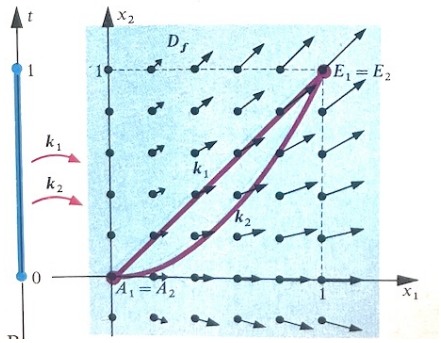
The following solution parametrizes x and y with parameter t so that the endpoint $(2,2)$ corresponds to $t=1$: $x=2t,\ y=2t,\ g=(x,y),\ F=(y^2,0)$. Then we get

```
load_package avector;
g := avec(2*t_,2*t_,0);
F := avec((2*t_)^2,0,0);
defint( F dot df(g, t_), t_, 0, 1);
```

$\frac{8}{3}$

Get *code*.

Exercise 71. (from [38, p.350])



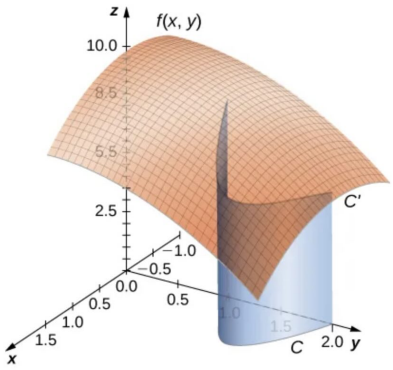
Line integral along two different paths.

Let $f : D \to \mathbb{R}^2$ be a vector field defined by $(x_1, x_2) \mapsto (x_1, x_1 \cdot x_2)$.
Let $k_1 : (0,1) \to \mathbb{R}^2$ be the line defined by $t \mapsto (t,t)$.
Let $k_2 : (0,1) \to \mathbb{R}^2$ be a curve defined by $t \mapsto (t,t^2)$.

Verify:
a. $\int_{k_1} f = \frac{5}{6}$
a. $\int_{k_2} f = 0.9$

[40] The following line integral problem is from *Advanced Calculus, Fifth Edition* by Wilfred Kaplan.

Exercise 72. (from OPENSTAX : line integral)



Redo the following examples resp. exercises from OPENSTAX using classical and differential forms methods:

- scalar field line integrals

a. Example 6.15
b. Example 6.17
c. Example 6.17
d. Example 6.17

- vector field line integrals

e. Example 6.19
f. Example 6.20
g. Example 6.21

12.4 Surface area

Fact 13. (Theoretical minimum) Let S be a surface parameterized by x and y. That is, let $S = (x, y, z)$ where $z = f(x, y)$. The tangent vectors at a point on S form a tiny parallelogram. The area a of this parallelogram is given by the magnitude of the cross product:

$$a = \left| \frac{\partial S}{\partial x} \times \frac{\partial S}{\partial y} \right| \tag{12.3}$$

By summing over all these parallelograms we obtain the total surface area A. Hence[41]

$$A = \iint dA = \iint a \, dx \, dy \tag{12.4}$$

Mental image

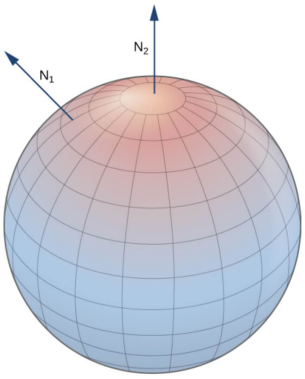


Figure 15: Surface of a sphere $S^2 \subset \mathbb{R}^3$ with tiny parallelogram meshes and orientation via outward oriented normal vectors. [picture from Lumen [30]]

Example 40. (cf. [49, p.26])
Compute the surface area of a unit disk parallel to the xy plane.

Solution

1st classic vector analysis

```
load_package avector;
z := 2;
S := avec(x, y, z);                    % S = (x,y, z)
df(S,x) cross  df(S,y);                % this is (12.3)
a := vmod( df(S,x) cross  df(S,y) );   % the norm is named vmod in Avector

defint( defint(a, y,-sqrt(1 - x^2),sqrt(1 - x^2)), x,-1,1);
```

[41] for more information see WIKIPEDIA Surface Area

The result is π, the area of a unit circle, which is what we expect.

Get *code*.

2nd using exterior calculus of differential forms

We pull back the standard area 2-form $\omega = dx \wedge dy$ into polar coordinates.
When $dx \wedge dy$ is executed, REDUCE`excalc` automatically applies the exterior derivative operator d to the coordinate definitions using the chain rule: $dx = \cos\theta\, dr - r \sin\theta\, d\theta$, $\quad dy = \sin\theta\, dr + r\cos\theta\, d\theta$.
Using the anti-commutative property of the wedge product where $dr \wedge dr = 0$ and $d\theta \wedge dr = -dr \wedge d\theta$, `excalc` simplifies $\omega = dx \wedge dy$ to $(\cos^2\theta + \sin^2\theta)\, r\, (dr \wedge d\theta) = r\, (dr \wedge d\theta)$.
The coframe and frame statements define a set of basis 1-forms $(dr, d\theta)$ and their corresponding dual tangent vector fields (e_r, e_θ). Contracting the 2-form ω twice cancels the differential forms to isolate the scalar component: $e_\theta \lrcorner (e_r \lrcorner (r\, dr \wedge d\theta)) = e_\theta \lrcorner (r\, d\theta) = r$
Ultimately, the `int` operator evaluates the double definite integral: $\int_0^{2\pi} \int_0^1 r\, dr\, d\theta = \pi$.

```
load_package excalc$

% Declare coordinates/parameters as 0-forms (scalars)
pform r=0, theta=0, c=0$

% Define the surface parametrization (z = 2 creates the parallel plane)
x := r * cos(theta);
y := r * sin(theta);
z := 2;

% Compute the area 2-form via exterior derivative and wedge product
area_form := d x ^ d y;

% set up orthonormal basis frame to contract and extract the integrand
coframe o(1) = d r, o(2) = d theta;
frame e;

% Extract the scalar coefficient using interior product contraction _|
integrand := e(2) _| (e(1) _| area_form);

% Integrate over the unit disk bounds
surface_area := int(int(integrand, r, 0, 1), theta, 0, 2*pi);
```

π

Get *code*.

Example 41.
Compute the surface area of $z = x^2 + 2y$ over a unit square.

Solution

1st using classic vector analysis using (12.4)

```
load_package avector;
S := avec(x, y, x^2 + 2*y);   % S: z = x^2 + 2y;
a := vmod( df(S,x) cross  df(S,y) );  %  abs is named vmod in Avector

area := defint( defint( a, x, 0,1), y,0,1);

on rounded;
print_precision 5;
area;
```

$\frac{5}{8}\log(5) + \frac{3}{2} \approx 2.5059$

Get *code*.

2nd using exterior calculus of differential forms I – explicit 2D function

We interpret the relation $z = x^2 + 2y$ as the 2D function $z(x, y) = x^2 + 2y$ over the unit square $[0, 1] \times [0, 1]$ as a 2D manifold. We then build the induced metric on this surface by $ds^2 = dx^2 + dy^2 + dz^2$ with $dz = 2x\ dx + 2\ dy$, i.e. $ds^2 = dx^2 + dy^2 + d(2x\,dx + 2\,dy)^2$ as line element and calculate the area form as the Hodge dual of the volume element '1' in the ambient space.– Alternatively, after the setup of the coframe with its metric, we have automatically pre-calculated the determinant of the metric g available under the name `detm!*` (= det g) and the whole calculation reduces to a one-liner.

```
load_package excalc;

% coordinates of unit square
pform {x,y} = 0;

% basis 1-forms and induced metric
coframe e(x) = d x, e(y) = d y
   with metric g = (1+4*x**2)*e(x)*e(x)
                 + 4*x*e(x)*e(y) + 4*x*e(y)*e(x)
                 + 5*e(y)*e(y);

pform area = 2;
area := # 1;

% Integrate the area coefficient sqrt(4*x2+5) over the unit square
coeff := coeffn( area, e(x)^e(y), 1);
a := int(int(coeff, x, 0, 1), y, 0, 1);
```

```
on rounded;
a;

% alternative in one step
a := int(int( sqrt(detm!*), x, 0, 1), y, 0, 1);
```

2.50590

Get *code*.

3rd using exterior calculus of differential forms II – implicit 3D function

We interpret the relation $z = x^2 + 2y$ as the 2D zero-level set $f^{-1}(0)$ of the scalar function $f(x, y, z) = z - x^2 - 2y \in \mathbb{R}$. The `coframe` command defines a basis of orthonormal 1-forms $e^1 = dx, e^2 = dy, e^3 = dz$, representing the standard Euclidean metric $ds^2 = (e^1)^2 + (e^2)^2 + (e^3)^2$ in the ambient space $\mathbb{R}^3$. This also activates the Hodge star operator `#`. The differential $df = 1dz - 2x\,dx - 2\,dy$ of f acts as the normal 1-form. Its Hodge dual `#df` then pulls this back to a 2-form on the surface. The norm $\|df\| = \sqrt{5 + 4x^2}$ is calculated via the identity `#(df ∧ #df)` $= \|df\|^2$, its square root is the integrand.

```
load_package excalc;

% Define the coordinates of the 3D ambient space
pform {x, y, z}=0;

% Establish an orthonormal coframe (Euclidean basis 1-forms)
coframe e(1) = d x, e(2) = d y, e(3) = d z;

% Define the surface implicitly as f(x,y,z) = 0
pform f=0, df=1;
f := z - x**2 - 2*y;
df := d f;

% Compute the norm squared of the normal vector field via Hodge star
pform norm_sq=0;
norm_sq := #(df ^ # df);

% integrate the square root
surface_area := int(int( sqrt(norm_sq),  y, 0,1), x, 0,1);
```

Get *code*.

Example 42.
Find the area of the spiral ramp (helicoid) defined by[42]

$$S = \begin{pmatrix} u\cos v \\ u\sin v \\ v \end{pmatrix}, \quad 0 \le u \le 1, \quad 0 \le v \le 3\pi$$

Solution

1st using classic vector analysis with (12.4)

```
load_package avector;

S := avec(u*cos(v), u*sin(v), v);    % x = u cos(v), y = u sin(v), z = v
a := vmod( df(S,u) cross  df(S,v) );
area := defint( defint(a, u,0,1),  v,0,3*pi);

on rounded;
print_precision 5;
area;
```

$$\frac{3\pi}{2^{1/2}} + \tfrac{3}{2}\pi \log\left(2^{1/2}+1\right)$$

10.8177

Get *code*.

2nd using exterior calculus of differential forms

On a 2-surface embedded in 3D Euclidean space, the induced area 2-form is calculated by the pullback of the Euclidean volume metric components: $\sqrt{(dx\wedge dy)^2 + (dy\wedge dz)^2 + (dz\wedge dx)^2}$.

```
% Load the differential geometry package
load_package excalc$

for all x let cos(x)**2 + sin(x)**2 = 1;

% Declare parameters u and v as 0-forms
pform u=0, v=0$

% Define the surface parametrization
x := u * sin(v);
y := u * cos(v);
z := v;

% Compute the standard coordinate differential forms
```

[42]This exercise is from *Multivariable Mathematics* by Williamson and Trotter, p. 598.

```
pform dx=1, dy=1, dz=1$
dx := d x;  dy := d y;  dz := d z;

% Set up a 2D basis frame o() to extract scalar coefficients
coframe o(1) = d u, o(2) = d v;
frame e;

% Extract the scalar coefficients from each 2-form component
f_xy := e(2) _| (e(1) _| (dx ^ dy));   % brackets important!
f_yz := e(2) _| (e(1) _| (dy ^ dz));
f_zx := e(2) _| (e(1) _| (dz ^ dx));

% integrand is the Euclidean norm of the pulled back wedge products
integrand := sqrt(f_xy**2 + f_yz**2 + f_zx**2);

% Integrate over the bounds: u from 0 to 1,  v from 0 to 3*pi
surface_area := int(int(integrand, u, 0, 1), v, 0, 3*pi);
```

$\frac{3\pi}{2}\left(\sqrt{2}+\ln(1+\sqrt{2})\right)$

REDUCE output:

```
f_xy := e(2) _| (e(1) _| (dx ^ dy));
```

$$\text{f_xy}:= -u$$

```
f_yz := e(2) _| (e(1) _| (dy ^ dz));
```

$$\text{f_yz}:= \cos(v)$$

```
f_zx := e(2) _| (e(1) _| (dz ^ dx));
```

$$\text{f_zx}:= -\sin(v)$$

```
integrand := sqrt(f_xy**2 + f_yz**2 + f_zx**2);
```

$$\text{integrand}:= \sqrt{u^2+1}$$

```
surface_area := int(int(integrand, u, 0, 1), v, 0, 3*pi);
```

$$\text{surface_area}:= \frac{3\pi\left(\sqrt{2}+\log\left(\sqrt{2}+1\right)\right)}{2}$$

 Get *code*.

12.4.1 Exercises



We do some selected examples and exercises from [11] with REDUCE`excalc`.
The reader may consult the book by DINEEN for the mathematical considerations.

Exercise 73. (Surface S of revolution, cf. [11, p.122 ff])
Let $P : [a, b] \to \mathbb{R}, t \mapsto P(t) = (x(t), y(t))$ be a parameterization of a curve to be revolved. The parameterization Φ of the surface of revolution S is defined by

$$\Phi(t, \theta) := (x(t), y(t) \cdot \cos\theta, y(t) \cdot \sin\theta) \tag{12.5}$$

on the open set $U := \{(t, \theta) : a < t < b, 0 < \theta < 2\pi\}$.

a. Verify: $\text{Area}(S) = 2\pi \int_a^b y(t) \cdot \|P'(t)\| dt$.
b. Verify: $\text{Area}(Cone_h_r) = \pi r(h^2 + r^2)^{1/2}$.
c. Verify: $\text{Area}(Cylinder_r_h) = 2\pi rh$.
d. Verify: $\text{Area}(Sphere_r) = 4\pi r^2$.
e. Verify: $\text{Area}(Torus_r_b) = 4\pi^2 rb$.
Use the parameterization $\Phi(\theta, \psi) = ((b + r\cos\theta)\cos\psi, (b + r\cos\theta)\sin\psi, r\sin\theta)$

Exercise 74. (from [11, p.130]))
Calculate the surface area of the paraboloid $z = x^2 + y^2$ which lies between the planes $z = 0$ and $z = 4$. $\frac{\pi}{6}(17^{3/2} - 1)$

Exercise 75. (from [11, p.130]))
Calculate the area of the helicoid the parameterization $P(r, \theta) = (\cos\theta, r\sin\theta, \theta)$ with $0 < \theta < 2\pi$ and $0 < r < 1$.

Exercise 76. (from [11, p.130]))
Calculate the surface area of the paraboloid $z = x^2 + y^2$which lies between the planes z=0 and z=4.

12.5 Surface integral

Fact 14. (Theoretical minimum) A surface integral is like adding up all the wind on a sail. In other words, we want to compute

$$\iint \mathbf{F} \cdot \mathbf{n}\, dA = ?$$

where $\mathbf{F} \cdot \mathbf{n}$ is the amount of wind normal to a tiny parallelogram dA. The integral sums over the entire area of the sail. Let S be the surface of the sail parameterized by x and y. (In this model, the z direction points downwind.) By the properties of the cross product we have the following for the unit normal $\mathbf{n}$ and for dA.

$$\mathbf{n} = \frac{\frac{\partial S}{\partial x} \times \frac{\partial S}{\partial y}}{\left| \frac{\partial S}{\partial x} \times \frac{\partial S}{\partial y} \right|} \qquad dA = \left| \frac{\partial S}{\partial x} \times \frac{\partial S}{\partial y} \right| dx\, dy$$

Hence

$$\iint \mathbf{F} \cdot \mathbf{n}\, dA = \iint \mathbf{F} \cdot \left(\frac{\partial S}{\partial x} \times \frac{\partial S}{\partial y} \right) dx\, dy \tag{12.6}$$

Mental image

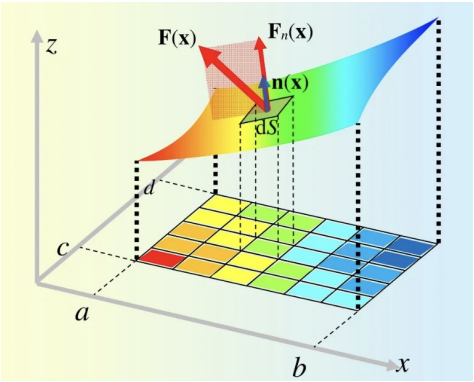


Figure 16: Surface integral on manifold S with normal $\mathbf{n}$ and area element dS.
[picture from youTube]

Example 43. Evaluate the surface integral $\iint_S \mathbf{F}\cdot\mathbf{n}\,d\sigma$ where $\mathbf{F} = xy^2z\mathbf{i} - 2x^3\mathbf{j} + yz^2\mathbf{k}$, S is the surface $z = 1 - x^2 - y^2$, $x^2 + y^2 \le 1$ and $\mathbf{n}$ is pointing upwards.[43]

Solution

- Note that the surface intersects the xy plane in a circle. By the right hand rule, crossing x into y yields $\mathbf{n}$ pointing upwards hence $\mathbf{n}\,d\sigma = (\partial_x S \times \partial_y S)\,dx\,dy$.
- The following code computes the surface integral. The symbol h is used as temporary variable. In the defining terms $F := (xy^2z, -2x^3, yz^2)$ and $S := (x, y, 1 - x^2 - y^2)$ we substitute $z := 1 - x^2 - y^2$.

1st using classic vector analysis using (12.6)

```
load_package avector;

F := avec(x*y^2*(1-x^2-y^2), -2*x^3, y*(1-x^2-y^2)^2);
S := avec(x,y,  1 - x^2 - y^2);
A := F dot (df(S,x) cross  df(S,y)) ;
h := sqrt(1 - x^2);

defint(defint(A, y, -h, h), x, -1, 1);

on rounded;
print_precision 5;
area;
```

$\frac{1}{48}\pi$

Get *code*.

2nd using exterior calculus of differential forms

We map the vector field to its corresponding 2-form, pull it back to the domain of the parameters, and integrate. Thereby REDUCE$^{\texttt{excalc}}$ expands and simplifies the exterior wedges automatically using anti-commutativity rules, and constructs the pullback coefficient

$$y(-4x^3 + 2x^2y(-x^2 - y^2 + 1) + (x^2 + y^2 - 1))^2(dx \wedge dy)$$

before we use REDUCE's integration command `int` over the domain disk.

```
load_package excalc;

pform {x,y,z, fx,fy,fz, c,c1}=0, w=2;

% component functions of vector field F
fx := x*y**2*z;
fy := -2*x**3;
```

[43] This exercise is from *Advanced Calculus* by Wilfred Kaplan, p. 313.

```
fz := y*z**2;

% define surface constrained, so d z becomes -2x d x - 2y d y
z := 1 - x**2 - y**2;

% define vector field F as 2-form w  and automatically
% evaluate the 2-form on the surface constraint

w := fx * d y^d z + fy * d z^d x + fz * d x^d y;

% capture coefficient of d x^d y with inner product
c := @(y) _| (@(x) _| w);

% convert to polar coordinates (dx^dy = r*dr^dth)
% and integrate with double integral over the unit disk x^2 + y^2 <= 1

c1 := r * sub( x=r*cos(th), y=r*sin(th), c);
int(int(c1, th, 0, 2*pi), r, 0, 1);
```

REDUCE output:

```
% evaluate the 2-form on the surface constraint
w := fx * d y^d z + fy * d z^d x + fz * d x^d y;
```

$$w:= d\,x\wedge d\,y\,y\left(-2x^4y+x^4-4x^3-2x^2y^3+2x^2y^2+2x^2y-2x^2+y^4-2y^2+1\right)$$

```
% capture coefficient of d x^d y with inner product
c := @(y) _| (@(x) _| w);
```

$$c:= y\left(-2x^4y+x^4-4x^3-2x^2y^3+2x^2y^2+2x^2y-2x^2+y^4-2y^2+1\right)$$

```
c1 := r * sub( x=r*cos(th), y=r*sin(th), c);
```

$$c_1:=$$
$$\sin(\mathrm{th})\,r^2$$
$$\left(-2\cos(\mathrm{th})^4\sin(\mathrm{th})\,r^5+\cos(\mathrm{th})^4r^4-4\cos(\mathrm{th})^3r^3-2\cos(\mathrm{th})^2\sin(\mathrm{th})^3r^5+2\cos(\mathrm{th})^2\sin(\mathrm{th})^2r^4+\right.$$
$$\left.2\cos(\mathrm{th})^2\sin(\mathrm{th})\,r^3-2\cos(\mathrm{th})^2r^2+\sin(\mathrm{th})^4r^4-2\sin(\mathrm{th})^2r^2+1\right)$$

```
int(int(c1, th, 0, 2*pi), r, 0, 1);
```

$$\frac{\pi}{48}$$

Get *code*.

- It is possible to do the double integral in two distinct steps:

```
radial    := int(r**5*(1-r**2),r,0,1);
angular   := int(cos(theta)**2*sin(theta)**2, theta,0,2*pi);
result    := 2*radial*angular;
```

12.5.1 Exercises



We recommend some selected examples and exercises from [34] with REDUCE`excalc`. The reader may consult the book by *openSTAX* for mathematical considerations.

Exercise 77. (surface integrals I)
Do some of the examples in *openSTAX*:
a. example 5.1
b. example 5.3
c. example 5.7
d. example 5.9

Use package `excalc` for the solution.

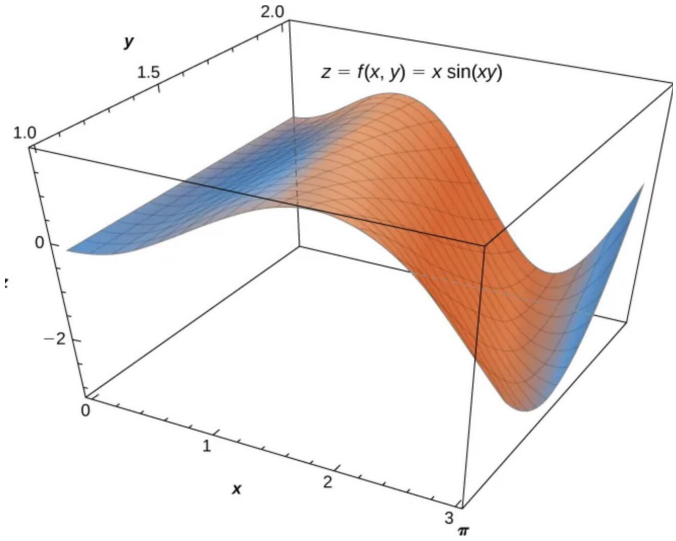


Figure to example 5.7 by *openSTAX*

Exercise 78. (surface integrals II)
Do some of the exercises in *openSTAX*
a. exercise 5.1
b. exercise 5.3
c. exercise 5.7
d. exercise 5.9

Use package `excalc` for the solution.

12.6 Volume integral

Fact 15. (Theoretical minimum) The '*volume integral*' is the triple integral

$$vol(D) := \iiint_D f(x,y,z)\,dV. \tag{12.7}$$

with the differential element $dV = dx\,dy\,dz$ within a region $D \subset \mathbb{R}^3$ of a function $f(x,y,z)$, usually written as $\iiint_D f(x,y,z)\,dx\,dy\,dz$.

Fact 16. The triple integral can be transformed from Cartesian coordinates to any arbitrary coordinate system using the determinant of the Jacobian matrix. If we have a transformation of coordinates from $(x,y,z) \mapsto (u,v,w)$, then the above integral is

$$\iiint_D f(x,y,z)\,dx\,dy\,dz = \iiint_D f(u,v,w) \left| \frac{\partial(x,y,z)}{\partial(u,v,w)} \right| du\,dv\,dw \tag{12.8}$$

Mental image

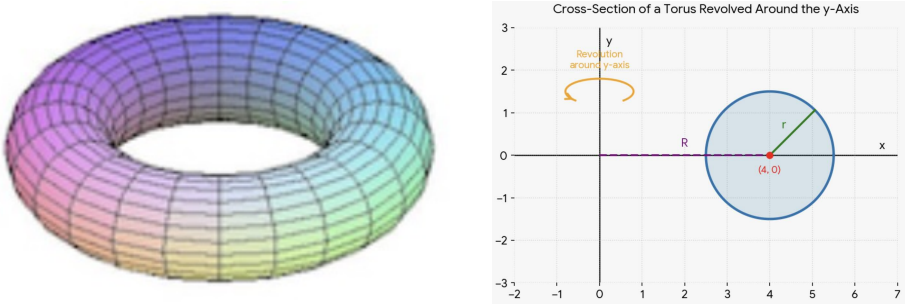


Figure 17: The torus $T : (x-R)^2 + y^2 = r^2$ as 2D manifold in ambient space $\mathbb{R}^3$ with $vol(T) = 2\pi^2 R r^2$

Example 44. Let M be the unit cube $[0,1]\times[0,1]\times[0,1]$ and let f be the density function $f(x,y,z) := x^2 + y^2 + z^2$. Calculate the mass of M.

Solution

1st using classic vector analysis using (12.7)

```
% I.  USING core Reduce
f:=1;
int( int( int(f ,x,0,1), y,0,1), z,0,1);  % triple integral over unit cube

f:=x^2+y^2+z^2;
int( int( int(f ,x,0,1), y,0,1), z,0,1);
```

```
% II.  USING Reduce package AVECTOR
load_package Avector;

vlb := avec(0,0,0);                % Lower bounds
vub := avec(1,1,1);                % Upper bounds
volintorder := avec(0,1,2);        % The order of integration x->y->z
volintegral(1,vlb,vub);
```

1

Get *code*.

2nd using exterior calculus of differential forms with REDUCE$^{\texttt{excalc}}$

This is only a tiny exercise in `excalc` syntax.

```
load_package excalc;

% Define x, y, z as coordinates
pform x=0, y=0, z=0;

% Define the density function
pform f=0;
f := x**2 + y**2 + z**2;

% Construct the volume 3-form using the wedge product (^)
pform omega=3;
omega := f * (d x ^ d y ^ d z);

% Extract the scalar coefficient of the 3-form omega
coeff := omega / (d x ^ d y ^ d z);

% Perform the sequential triple integration over the boundaries [0,1]
% in three consecutive steps and in one step

I1   := int(coeff, x, 0, 1);
I2   := int(I1, y, 0, 1);
I3   := int(I2, z, 0, 1);

int(int(int( coeff, x, 0, 1), y, 0, 1), z, 0, 1);
```

1

Get *code*.

Example 45. (WIKIPEDIA[44]) If we have a scalar density function on the unit cube then the volume integral will give the total mass of the cube.
For example, for density function $f : \mathbb{R}^3 \to \mathbb{R}, (x, y, z) \mapsto x + y + z$ the total mass of the cube is $mass = \int_0^1 \int_0^1 \int_0^1 (x + y + z)\, dx\, dy\, dz = \int_0^1 \int_0^1 \left(\frac{1}{2} + y + z\right) dy\, dz = \int_0^1 (1 + z)\, dz = \frac{3}{2}$

Solution with REDUCE:

```
f := x+y+z;
int( int( int(f, x,0,1), y,0,1), z,0,1);     % triple integral
```

$\frac{3}{2}$

○ A solution with REDUCE[excalc] goes analog to example.44.

Get *code*.

Example 46. Study both examples of triple integrals given in package `avector`.

For your convenience, we reproduce the second one here. – We wish to calculate the volume of a right circular cone. This is equivalent to integrating unity over a conical region with the bounds: $z : \int_0^h, r : \int_0^{p*z}, \phi : \int_0^{2*pi}$, where $p = ratio\ of\ base\ diameter\ to\ height$.
We evaluate the volume by integrating a series of infinitesimally circular disks of constant z-value. The integration is thus performed in the order: $d\phi$ from 0 to 2π, dr from 0 to pz, dz from 0 to h. The order of the indices is thus 2, 0, 1.

Solution

```
load_package Avector;

volintorder := avec(2, 0, 1);
vlb := avec(0,   0, 0);
vub := avec(p*z, h, 2*pi);
volintegral(1, vlb, vub);
```

REDUCE output:

```
vub := avec(p*z, h, 2*pi);
```

$$\begin{aligned} \mathrm{vec}(x) &:= p\, z \\ \mathrm{vec}(y) &:= h \\ \mathrm{vec}(z) &:= 2\,\pi \end{aligned}$$

```
volintegral(1, vlb, vub);
```

$$2\, h\, p\, \pi\, z$$

$2hp\pi z = 2r\pi z$

Get *code*.

[44] See Volume integral

12.6.1 Exercises

Exercise 79. (volume of torus)
Verify the formula $vol(T) = 2\pi^2 Rr^2$ for the volume of the torus $T : (x - R)^2 + y^2 = r^2$, cf. fig.17. using
a. Method 1: washer method (integrating along y)
b. Method 2: cylindrical shells method (integrating along x)



Do some selected examples and exercises from *openSTAX* with REDUCE[excalc]. The reader may consult *openSTAX* for mathematical considerations.

Exercise 80. (volume integrals I)
Do some of the examples in *openSTAX*
a. example 5.38
b. example 5.39
c. example 5.41
d. example 5.45
e. example 5.50
f. example 5.52
g. example 5.53

Use package `excalc` for the solution.

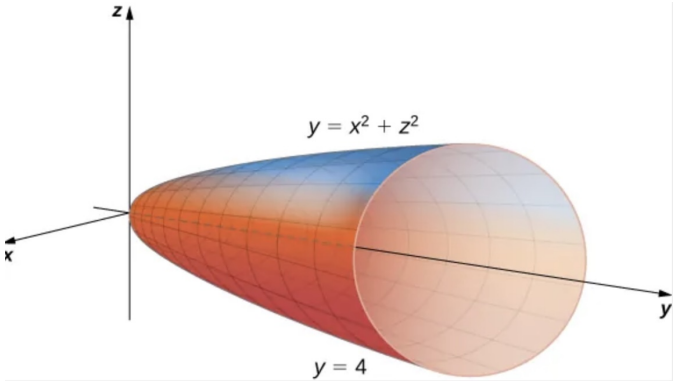


Figure to example 5.38 by *openSTAX*

Exercise 81. (volume integrals II)
Do some of the exercises in *openSTAX*
a. exercise 212
b. exercise 240
c. exercise 569

Use package `excalc` for the solution.

13 Generalised Stokes' Theorem

One of the most important corollaries of the theorem on exterior derivatives is the Newton-Leibniz-Gauss-Green-Ostrogradskii-Stokes-Poincare formula: cf. Arnold [3, p.192]

$$\int_M d\omega = \int_{\partial M} \omega = \int_D \Phi^* \omega \tag{13.1}$$

$$\overset{explicit}{:=} \int_D \omega(\partial_{u_1}\Phi, ..., \partial_{u_k}\Phi) du_1 \cdots du_n$$

where

- ω is a differentiable k-form
- M is a differentiable k-dimensional manifold
- $\Phi : D \subset \mathbb{R}^k \to \mathbb{R}^n$ with $(u_1, ..., u_k) \in D$, $k < n$ is a smooth parameterization e.g. a 1-form integrated along a curve in ambient space $\mathbb{R}^3$:

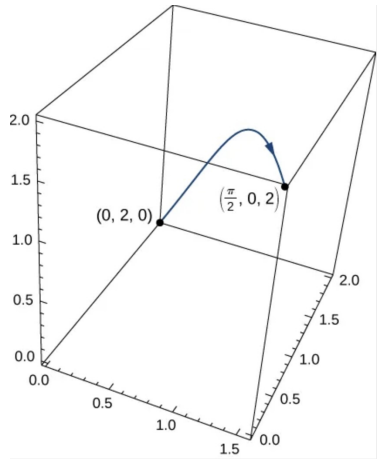


Figure to example 6.22 by *openSTAX*

In the following, we will show how this general Stokes' theorem relates back to the great 4 classical integral theorems of vector analysis: the gradient theorem, Green's theorem, Stokes' theorem and Gauss' theorem. And how these theorems are special cases of the unifying *generalized* Stokes' *theorem* (13.1).

13.1 Gradient Theorem

Fact 17. (Theoretical minimum) This is the gradient theorem:[45]

$$\oint_C grad(f) \bullet d\phi := \int_a^b grad(\phi(t)) \bullet \phi\,'(t)\, dt = f(B) - f(A) \tag{13.2}$$

where $f : U \subset \mathbb{R}^k \to \mathbb{R}$, $k \in \{2, 3\}$ is a smooth function and $\phi : [a, b] \to C$ is a parametrization of any closed curve C from $A := f(a)$ to $B := f(b)$.

- If we set $M := [a, b]$ with $\partial M = \{a, b\}$, $\omega := df = grad(f)$, we recognize formula (13.1).
- If $F = grad(f)$ with a certain f, then F is called *conservative* and (13.2) is fulfilled. [46]
- This theorem is also called the *Fundamental Theorem for Line Integrals*.

Mental image

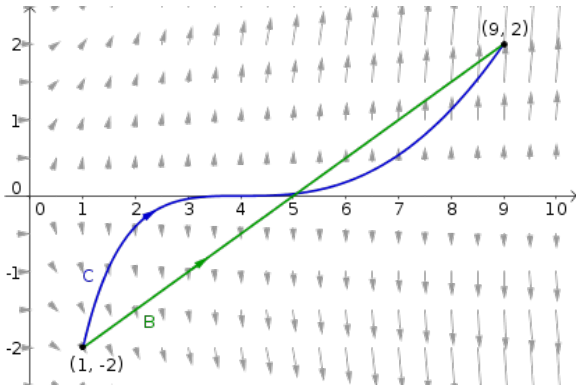


Figure 18: The line integral of a *gradient* or conservative vector field along a *closed* curve C depends only on the endpoints of the curve.
Two paths with endpoints $A = (1, -2)$ and $B = (9, 2)$ in $\mathbb{R}^2$ in a vector field.

.

Example 47. Let C be the curve defined by $\phi(t) := (t^9, \sin^9(\pi t/2))$, $t \in [0, 1]$.
Verify the gradient theorem for the integral $\int_C y\, dx + x\, dy$, cf. [31, p. 897]

Solution

1st using classic vector analysis using (12.7)

We calculate both sides of (13.2) and show their sameness.

[45] for more information, i.e. the prerequisites, see WIKIPEDIA Gradient Theorem
[46] See *openSTAX*

```
% I. we use core REDUCE
% CHECK LHS:

% Define the parametric equations of the curve C
x := t^9;
y := sin(t*pi/2)^9;

% Calculate the differential components dx/dt and dy/dt
% and do the dot product with the vector field components
% F * dr =  y*(dx/dt) + x*(dy/dt)
integrand := y * df(x, t) + x * df(y, t);

% Calculate the definite line integral from t = 0 to t = 1
result := int( integrand, t, 0, 1);

% CHECK RHS:
operator f;
f := x*y;
sub(x=1,y=1, f) - sub(x=0,y=0,f);   %  = f(1,1) - f(0,0);

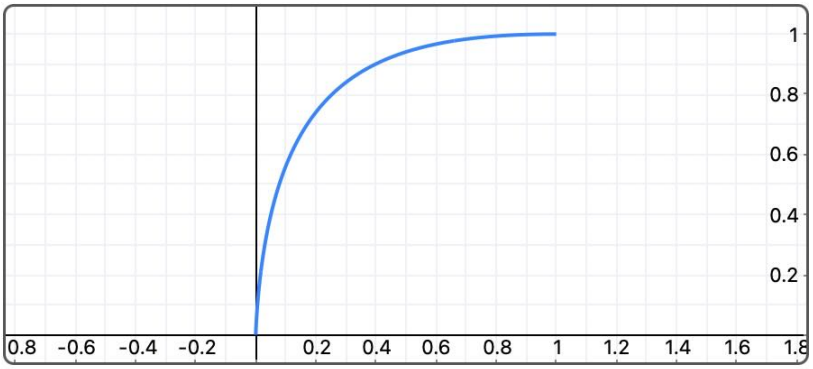
% II. We use package AVECTOR as alternative
load_package avector;

% Define 3D vector field F in Cartesian coordinates
F := avec(y, x, 0);

% Define the parameterized curve C using parameter t
C := avec(t**9, sin(t*pi/2)**9, 0);     % t*pi/2 ~ 1.5708*t

% Evaluate the definite line integral from 0 to 1
deflineint(F, C, t, 0, 1);
```

1 Get *code*.



Plot done with MATHSTUDIO.

```
ParametricPlot(u^9, sin(u/2*pi)^9, u=[0,1,400], color=blue)
```

2nd using exterior calculus of differential forms with REDUCE$^{\texttt{excalc}}$

We verify the LHS.

```
load_package excalc;

% Define t as the base manifold coordinate
coframe e(t) = d t;
pform {x,y}=0;

% Declare x and y as scalar functions dependent on t
fdomain x = x(t), y = y(t);

% Define components of parametrization phi(t)
x := t**9;
y := sin(t*pi/2)**9;

% Define the field-associate 1-form omega = y*dx + x*dy
pform omega = 1;
omega := y * d x + x * d y;

% Pull back the 1-form to the t-manifold.
% This automatically evaluates the exterior derivatives
% d x and d y via the chain rule.
omega;

% Extract the scalar integration coefficient by dividing by 'd t'
% or use coeffn(..) or substitute dt ->1 in omega
pullback1 := omega / d t;
pullback2 := coeffn(omega, d t, 1);
pullback3 := sub(d t = 1, omega);

% Perform the definite line integration from t = 0 to t = 1
result := int(pullback1, t, 0, 1);
```

REDUCE output: Get *code*.

omega;

$$\frac{9\,e^{t}\sin\left(\frac{\pi t}{2}\right)^{8}t^{8}\left(\cos\left(\frac{\pi t}{2}\right)\pi t+2\sin\left(\frac{\pi t}{2}\right)\right)}{2}$$

% Extract the scalar integration coefficient
pullback1 := omega / d t;

$$\text{pullback}_1 := \frac{9\sin\left(\frac{\pi t}{2}\right)^{8}t^{8}\left(\cos\left(\frac{\pi t}{2}\right)\pi t+2\sin\left(\frac{\pi t}{2}\right)\right)}{2}$$

13.1.1 Exercises



Do some selected examples and exercises from [35] with REDUCE`excalc`.
You should consult the book OPENSTAX for the mathematical considerations.

Exercise 82. Do some of the examples in *openSTAX* Ch. 6.2

a. example 6.28

b. example 6.29

c. example 6.26

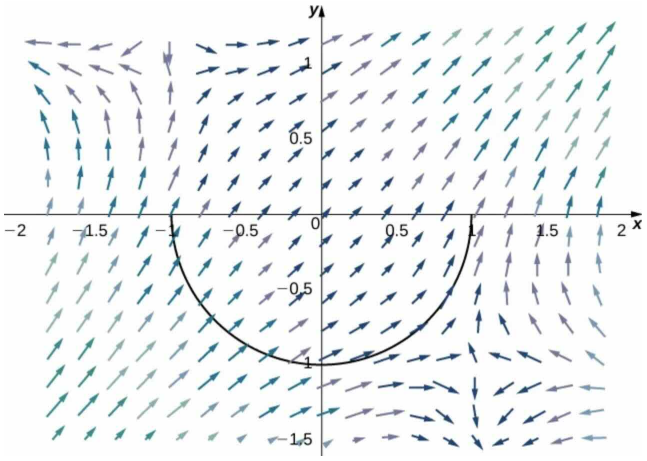


Figure to example 6.38 by *openSTAX*

d. Do checkpoint 6.32. Use package `excalc` for the solution.

13.2 Green's Theorem

Fact 18. (Theoretical minimum) This is Green's theorem in $\mathbb{R}^2$ in classical notation:[47]

$$\oint_C (P\,dx + Q\,dy) = \iint_D (\frac{\partial Q}{\partial x} - \frac{\partial P}{\partial y})dx\,dy \tag{13.3}$$

where $F = (P, Q) : \mathbb{R}^2 \to \mathbb{R}^2$ is a smooth vector field with component functions P and Q and C is a positiv-oriented simple curve enclosing some region D in the xy-plane $\mathbb{R}^2$.

Mental image

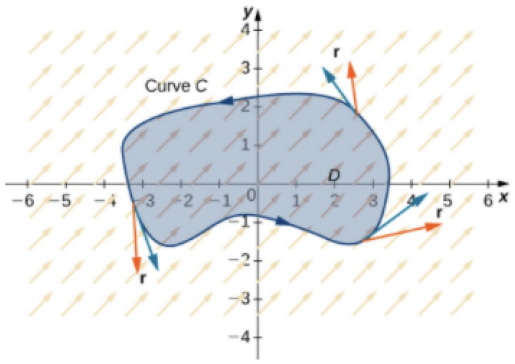


Figure 19: Green's theorem relates a line integral along a closed curve C to a double integral over the enclosed region D in $\mathbb{R}^2$. [picture from OPENSTAX Green]

Mathematical reflection on the equivalence of (13.3) with $\oint_{\partial M} \omega = \int_M d\omega$

We want to verify that Formula (13.3) is an incarnation of the generalized formula (13.1).
Let $F = (P, Q) \equiv P \cdot \vec{e_1} + Q \cdot \vec{e_2}$, where $\vec{e_1} = (1, 0), \vec{e_2} = (0, 1) \in \mathbb{R}^2$.
Translate F to its associated 1-form $\omega = P \cdot dx + Q \cdot dy \in \Omega^1\mathbb{R}^2$.
Then we can write

$$\int_{\partial M=C} \omega \overset{p.d.}{=} \int_C P \cdot dx + Q \cdot dy \overset{old\ notation}{=} \int_C F \bullet d\Phi$$
$$\overset{(13.1)}{=} \int_{M=D} d\omega = \int_D d(P \cdot dx + Q \cdot dy)$$

[47] for more information, i.e. the prerequisites, see WikiPediA Green's Theorem

$$
\begin{aligned}
&= \int_D \left(\frac{\partial P}{\partial y} dy \wedge dx + \frac{\partial Q}{\partial x} dx \wedge dy \right) \\
&= \int_D \left(\frac{\partial Q}{\partial x} - \frac{\partial P}{\partial y} \right) dx \wedge dy && \text{oriented area} \\
&= \iint_D \left(\frac{\partial Q}{\partial x} - \frac{\partial P}{\partial y} \right) dx\, dy && \text{unoriented area}
\end{aligned}
$$

□

Example 48. Evaluate $\oint_C (2x^3 - y^3)dx + (x^3 + y^3)dy$ around the circle $C : x^2 + y^2 = 1$ using Green's theorem. [48]

Solution

1. We calculate the RHS of (13.3).
Use polar coordinates to solve the surface integral, i.e. the area of circle C.

```
P := 2x^3 - y^3;
Q := x^3 + y^3;

f := df(Q,x) - df(P,y);
let {x=>r*cos(theta), y => r*sin(theta)};
let sin(x)^2+cos(x)^2   => 1;

int(int( f * r, r, 0, 1), theta, 0, 2*pi);
```

$\frac{3}{2}\pi$

- The integrand is $f \cdot r$ because $r\, dr\, d\theta = dx\, dy$.

Get *code*.

2. We calculate the LHS of (13.3).
We compute the line integral side of Green's theorem and see if we get the same result.

Solution

```
x := cos t_;
y := sin t_;
P := 2x^3 - y^3;
Q := x^3 + y^3;
f := P*df(x,t_) + Q*df(y,t_);
int(f, t_, 0, 2*pi);
```

$\frac{3}{2}\pi$

Get *code*.

[48] This exercise is from *Advanced Calculus* by Wilfred KAPLAN, p. 287.

3. We calculate the integral using differential forms with REDUCE[excalc].

```
load_package excalc;

% Declare coordinates as 0-forms and omega as 1-form
pform x=0, y=0, r=0, phi=0, omega=1;

% Define the differential form
omega := (2*x**3 - y**3)*d x + (x**3 + y**3)*d y;

% Compute the exterior derivative
%         which applies Green's Theorem automatically
d_omega := d omega;

% Set up the transformation to polar coordinates
x := r*cos(phi);
y := r*sin(phi);

% Evaluate d_omega w.r.t. polar coframe (dphi ^ dr)
polar_form := d_omega;

% Extract the coeff of d_omega and calculate the double integral
integrand := polar_form / (d phi ^ d r);
int(int( integrand, r,0,1), phi,0,2*pi);
```

Get *code*.

Example 49. Compute both sides of Green's theorem for the vector field $F = (1 - y, x)$ over the disk $x^2 + y^2 \leq 4$. – First compute the line integral along the boundary of the disk. Note that the radius of the disk is 2.

Solution

1. We calculate the LHS and RHS of (13.3) using core REDUCE.

```
% 1. ---------- Line integral
let {x=>2*cos(t_), y=>2*sin(t_)};    %  circle C: 2*(cos t, sin t) with r=2
P := 1 - y;
Q := x;
int( P*df(x,t_) + Q*df(y,t_), t_, 0, 2*pi);

% 2. ---------- Surface integral
P := 1 - y;
Q := x;
h := sqrt(4 - x^2);
f := df(Q,x) - df(P,y);
int( int( f, y, -h, h), x, -2, 2);
```

```
% 3. ---------- Surface integral: use polar coordinates
P := 1 - y;
Q := x;
f := df(Q,x) - df(P,y);
x := r*cos(theta);                          % change to polar coordinates
y := r*sin(theta);
int(int( f * r, r, 0,2), theta, 0,2*pi);    % integrate first over r
int(int( f * r, theta, 0,2*pi), r, 0,2) ;   % integrate first over theta
```

8π

Get *code*.

◦ In this case, REDUCE solved both forms of the polar integral. However, in cases where REDUCE fails to solve a double integral, try changing the order of integration.

2. We calculate the LHS and RHS of (13.3) using differential forms with REDUCE$^{\texttt{excalc}}$.

```
load_package excalc;

% Declare coordinates and forms
pform x=0, y=0, r=0, phi=0, omega=1;

% translate the vector field F to the associated 1-form
omega := (1 - y)*d x + x*d y;

% .......... LHS: Line Integral around the boundary C (r = 2)

% substitute boundary parameterization for a circle C of radius 2
% and pullback omega to the boundary parameter space with 1-dim
% coordinate phi
omega_C := sub({x = 2*cos(phi), y = 2*sin(phi)}, omega);

% Extract integrand for the line integral from 0 to 2*pi
line_integrand := omega_C / (d phi);
Int_C := int(line_integrand, phi, 0, 2*pi);

% .......... RHS: Area Integral over the disk D (r <= 2)

% Compute the exterior derivative of omega
dw := d omega;

% transform dw to polar coordinates for 2-dim area D
% and calculate (d omega) in terms of polar coordinates
% where automatically (dx^dy) -->  2*r*(dr^dphi)
```

```
omega_D := sub({x = r*cos(phi), y = r*sin(phi)}, dw);

% Extract integrand for the 2-dim integral with coordinates r and phi

area_integrand := omega_D /(d r ^ d phi);   % = 2*r;
Int_D := int(int(area_integrand, r, 0, 2), phi, 0, 2*pi);

% RESULTS:
write "LHS (Line Integral around boundary): ", Int_C;
write "RHS (Area Integral of disk): ", Int_D;

end;
```

Get *code*.

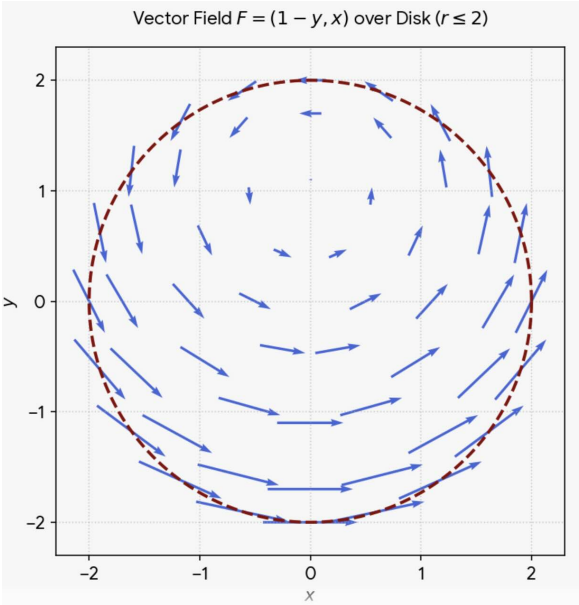


Figure from *openSTAX*

13.2.1 Exercises

Exercise 83. Redo example 48 with REDUCE package `avector`.

Exercise 84. Redo example 49 with REDUCE package `avector`.

Exercise 85. (from [11, p.96]))
Calculate the integral $I = \int_\Gamma (5 - xy - y^2)dx + (2xy - x^2)dy$ where Γ is the boundary of the unit square $[0,1] \times [0,1]$ in $\mathbb{R}^2$. $\frac{3}{2}$

Exercise 86. (from [11, p.98-9.8.b])
Verify Green's theorem for the integral $I = \int_\Gamma (x^2 + 2y^3)dy$, Γ is the circle $[(x-2)^1 + y^2 = 4$ in $\mathbb{R}^2$. 16π

Exercise 87. (from [31, p.911, E4]))
Let C be the boundary of unit square $[0,1] \times [0,1]$ oriented counterclockwise.
Evaluate the integral $\int_C (y^4 + x^3)dx + 2x^6 dy$ in $\mathbb{R}^2$. 1



Do some selected examples and exercises from Green Theorem with REDUCE[excalc].
You should consult the book OPENSTAX for the mathematical considerations.

Exercise 88 (Applying Green's Theorem over a Rectangle, Example 6.38)**.**
Calculate the line integral $\int_C x^2 y dx + (y - 3)dy$, where C is a rectangle with vertices $(1,1), (4,1), (4,5)$, and $(1,5)$ oriented counterclockwise.

Exercise 89 (Applying Green's Theorem to Calculate Work, Example 6.39)**.**
Calculate the work done on a particle by force field $F(x,y) = (y + \sin x, e^y - x)$ as the particle traverses circle $x^2 + y^2 = 4$ exactly once in the counterclockwise direction, starting and ending at point $(2,0)$.

Exercise 90. Do some of the examples in *openSTAX*
a. Example 6.40
b. Example 6.41
c. Example 6.42
d. Example 6.43

Exercise 91. Do some of the checkpoints in *openSTAX*
a. Checkpoint 6.34
b. Checkpoint 6.35

13.3 STOKE's Theorem

Fact 19. (Theoretical minimum) This is Stokes's theorem in Cartesian coordinates.[49]

$$\iint_S (\operatorname{curl} \mathbf{F}) \bullet \mathbf{n}\, dS = \oint_{\partial S} (P\, dx + Q\, dy + R\, dz) =: \int_C \mathbf{F} \bullet d\mathbf{r} \tag{13.4}$$

where $F = (P, Q, R) : \mathbb{R}^3 \to \mathbb{R}^3$ is a vector field with smooth component functions P, Q, R and $C := \partial S$ is a positiv-oriented curve enclosing an oriented surface S in the 3D space $\mathbb{R}^3$ with coordinates x, y, z. For surface S parametrized by x and y we have the surface normal vector $\mathbf{n}$ and the infinitesimal area element $dS = dx\, dy \overset{polar}{=} r\, dr\, d\phi = ..$:

$$\mathbf{n}\, dS = (\frac{\partial S}{\partial x} \times \frac{\partial S}{\partial y})\, dx\, dy \tag{13.5}$$

Mental image

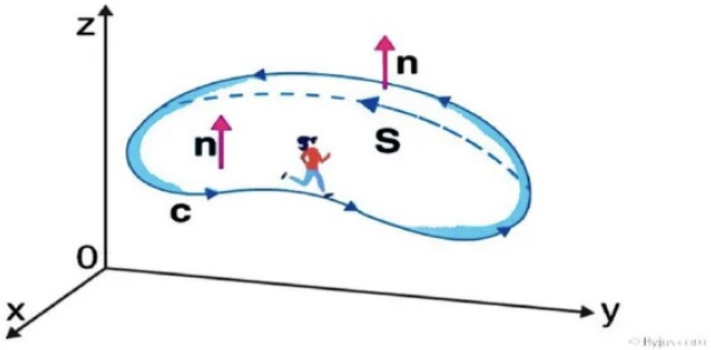


Figure 20: Stokes in words: the line integral of the tangential component of a vector field F around C is equal to the surface integral of the normal component of the curl of F on S.

Mathematical reflection on the equivalence of (13.4) with $\oint_{\partial M} \omega = \int_M d\omega$

We reproduce the argumentation in §13.2 in a 3D version in $\mathbb{R}^3$.
Let $F = (P, Q, R) \equiv P \cdot \vec{e_1} + Q \cdot \vec{e_2} + R \cdot \vec{e_3}$, where $\vec{e_1} = (1, 0, 0), ... \in \mathbb{R}^3$.
Translate F to its associated 1-form $\omega = P \cdot dx + Q \cdot dy + R \cdot dz \in \Omega^1\mathbb{R}^3$.

[49] for more information, i.e. the prerequisites, see WIKIPEDIA Stoke's Theorem

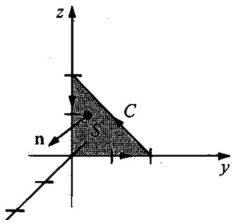
Then we can write

$$\begin{aligned}
\int_{\partial M=C} \omega &\overset{p.d.}{=} \int_C P\cdot dx + Q\cdot dy + R\cdot dz \overset{old\ notation}{=} \int_C F \bullet d\Phi \\
&\overset{(13.4)}{=} \int_{M=S} d\omega = \int_S d(P\cdot dx + Q\cdot dy + R\cdot dz) \\
&= \int_S \left(\tfrac{\partial R}{\partial y} - \tfrac{\partial Q}{\partial z}\right) dy\wedge dz + \int_S \left(\tfrac{\partial P}{\partial z} - \tfrac{\partial R}{\partial x}\right) dz\wedge dx + \int_S \left(\tfrac{\partial Q}{\partial x} - \tfrac{\partial P}{\partial y}\right) dx\wedge dy \\
&\ldots \quad \text{using} \quad dy\wedge dz \equiv \vec{e_1}, dz\wedge dx \equiv \vec{e_2}, dx\wedge dy \equiv \vec{e_3} \\
&= \iint_S (\operatorname{curl}\mathbf{F}) \bullet \mathbf{n}\, dS
\end{aligned}$$

□

First we will discuss how to interpret the LHS of the Stokes theorem (13.4).

Example 50 (Demystify the integrand $curl(\mathbf{F}) \bullet \mathbf{n}\, dS$)**.**
Calculate the integrand $curl(\mathbf{F}) \bullet \mathbf{n}\, dS$ of the vector field $F = (x^2, y^2, 1) \overset{0}{=} x^2\mathbf{i} + y^2\mathbf{j} + 1\mathbf{k}$, where the surface S is the triangle with vertices $(0,0,0), (0,2,0), (0,0,2)$, cf. [31, p. 920]. Program the *curl* operator using the REDUCE package `listvecops` to calculate the integrand.



We see: the surface normal is $(1,0,0)$

Solution

1st some remarks to the notations.

The term $curl(\mathbf{F}) \bullet \mathbf{n}\, dS$ consists of 4 parts: the *curl* of the vector field $\mathbf{F}$, the inner product $\bullet$, the surface normal $\mathbf{n}$ and the area element dS (which is only used for the integral calculation). Lets look at this process step-by-step.
1. For a three-dimensional vector field $\mathbf{F}(x,y,z) = f_1\mathbf{i} + f_2\mathbf{j} + f_3\mathbf{k}$, written as linear combination of the basis vectors $(\mathbf{i},\mathbf{j},\mathbf{k})$ of $\mathbb{R}^3$, the *curl* is defined as the cross product of the del operator ∇[50] and the vector field $\mathbf{F}$, denoted as $\nabla\times\mathbf{F}$ or $curl\,\mathbf{F}$:

$$\overset{curl\ F}{\nabla\times\mathbf{F}} \overset{1}{:=} \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ \frac{\partial}{\partial x} & \frac{\partial}{\partial y} & \frac{\partial}{\partial z} \\ f_1 & f_2 & f_3 \end{vmatrix} \overset{2}{:=} \left(\frac{\partial f_3}{\partial y} - \frac{f_2}{\partial z}\right)\mathbf{i} + \left(\frac{\partial f_1}{\partial z} - \frac{\partial f_3}{\partial x}\right)\mathbf{j} + \left(\frac{\partial f_2}{\partial x} - \frac{\partial f_1}{\partial y}\right)\mathbf{k} \tag{13.6}$$

0: to use formula (13.6), the vector field has to be in component notation with $\mathbf{i},\mathbf{j},\mathbf{k}$

[50] i.e. the gradient *grad*

1: $\nabla \times \mathbf{F}$ is only a *formal-symbolic mnemonic notation*, which will only remember to the phases of the calculation: do ∇ = *gradient*, then $\times$ with $\mathbf{F}$. It is a shortcut notation for the true definition RHS of (13.6).

The formal-symbolic 'determinant' places these ingredients in the rows of her matrix:
- the basis unit vectors $\mathbf{i}, \mathbf{j}, \mathbf{k}$ goes in row 1,
- the partial derivations operators $\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z}$ of the gradient vector ∇ goes into row 2
- the coordinate functions of the 3D vector field F are in row 3.

2: this is the true formula, which is the result of the Laplace-expansion of the determinant and is used for implementing *curl* in REDUCE.
The formal determinant is not programmable by a CAS – because it is 'formal-mental'.

2. The cross product $\mathbf{n}$ of two vectors $\mathbf{a} = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix}$ and $\mathbf{b} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}$ is defined by

$$\mathbf{a} \times \mathbf{b} := \begin{pmatrix} a_2b_3 - a_3b_2 \\ a_3b_1 - a_1b_3 \\ a_1b_2 - a_2b_1 \end{pmatrix} \overset{mnemonic}{=} \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \end{vmatrix} \tag{13.7}$$

which we also write formal symbolic as a *mnemonic device* 3×3 *'determinant'* .
Therefore only the vector term in the middle of (13.7) is programmable.
This is done next.

2nd realizing *curl* and *cross* in REDUCE.

We program the definitions for *curl* and *cross* as operators using package `listvecops`. In `listvecops` we represent 3D vectors as lists {..} and program procedures with the declaration `listproc`. To catch the n^{th} component of a vector we use the operator '`_n`'.

```
load_package listvecops;

listproc cross(a, b);  {(a _2)*(b _3) - (a _3)*(b _2),
                        (a _3)*(b _1) - (a _1)*(b _3),
                        (a _1)*(b _2) - (a _2)*(b _1)};

cross({1,2,3}, {4,5,6});

listproc curl(F);  {df(F _3,y) - df(F _2,z),
                    df(F _1,z) - df(F _3,x),
                    df(F _2,x) - df(F _1,y)};

curl({x^2,y^2,1});    % = {0,0,0} ok.

F:= {-y^3,x^3,-z^3};
curl(F);              % = {0,0, 3x^2+3y^2}
```

```
listproc normalize v;          v / sqrt(v *. v);
define normal = cross;
listproc unit_normal (a,b);    normalize( normal(a,b) );

% vertices
A := {0,0,0};  B := {0,2,0};  C := {0,0,2};
N := unit_normal (B-A,C-A);
```

$n = \{1, 0, 0\}$

Get *code*.

Example 51. Let $F = (y, z, x) : \mathbb{R}^3 \to \mathbb{R}^3$ be a vector field and let S be the surface of the paraboloid $z = 4 - x^2 - y^2$ above the xy plane. The intersection of the paraboloid and the xy plane is the circle $x^2 + y^2 = 4$. The following script computes both the surface and line integrals. Polar coordinates are used for the line integral so that operator *int* can succeed. – Verify Stokes' formula using REDUCE and package `avector`.

Solution

1st using classic vector analysis

```
% 1. ---------- LHS of Stoke's formula (13.4) using avector
load_package avector;

F := avec(y, 4-x^2-y^2, x);          % substituted z = 4 - x^2 - y^2
S := avec(x, y, 4-x^2-y^2);

A := curl(F) dot (df(S,x) cross df(S,y)) ;

x := r*cos(theta);
y := r*sin(theta);

defint( defint( A * r, r, 0, 2), theta, 0, 2*pi);

% 2. ---------- RHS of Stoke's formula (13.4) using core Reduce

x := 2*cos(t_);                      % parametrize circle with r=2
y := 2*sin(t_);
z := 0;
P := y;
Q := 4 - x^2 - y^2;                  % z = 4-x^2-y^2
R := x;
f := P*df(x,t_) + Q*df(y,t_) + R*df(z,t_);
int(f, t_, 0, 2*pi);
```

-4π

Get *code*.

▷ solution using a new version of `avector`

```
% I. verification of Stoke's formula (13.4) using new Avector
%    contributed by F.J. Wright
%    a very compact solution using an enhanced version of Avector

F := avec(y, z, x);

surf_int := surfint(curl F, avec(x, y, 4 - x^2 - y^2),
                x, -sqrt(4-y^2), sqrt(4-y^2),  y,-2,2);

line_int := lineint(F, avec(x, sqrt(4-x^2), 0), x, 2,-2)
            + lineint(F, avec(x, -sqrt(4-x^2), 0), x, -2,2);

% II. verification of Stoke's formula (13.4) using Avector
%     with new limits notation contributed by F.J. Wright

F := avec(y, z, x);

surf_int := surfint(curl F, avec(x, y, 4 - x^2 - y^2),
                x = -sqrt(4-y^2) .. sqrt(4-y^2), y = -2 .. 2);

line_int := lineint(F, avec(x,  sqrt(4-x^2), 0), x = 2 .. -2) +
            lineint(F, avec(x, -sqrt(4-x^2), 0), x = -2 .. 2);
```

-4π

Get *code*.

2nd using exterior calculus of differential forms with REDUCE$^{\texttt{excalc}}$

```
load_package excalc;

% declare coordinates, integrand (c) and omega (w) as forms
pform   x=0, y=0, z=0, c=0,          w=1, dw=2;

% Surface S: z = 4 - x^2 - y^2, parametrised by (x,y)
z := 4 - x**2 - y**2;

% omega = F . dr   [z is substituted, so dz and dw are
% automatically pulled back to the (x,y)-plane]
w  := y*d x + z*d y + x*d z;
dw := d w;

% We see: dw = (-2x-2y-1)*dx^dy with coeff c := (-2x-2y-1)
% we extract the integrand c e.g. with interior products
c := @(y) _| (@(x) _| dw);

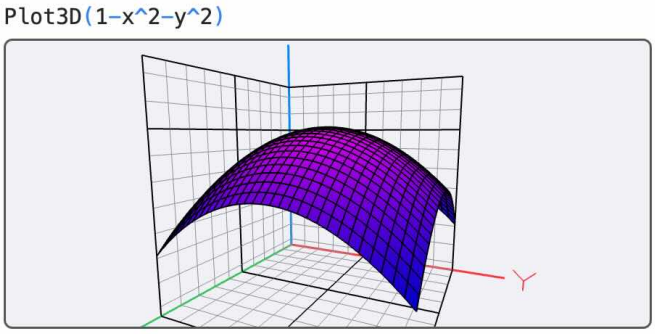
% LHS: Surface integral over the disk x^2+y^2 <= 4
%      use polar coordinates and dA = r * dr dt
surface_int := int(int( sub(x=r*cos(t), y=r*sin(t), c) * r,
                        r, 0,2), t, 0,2*pi);

% RHS: Line integral over the boundary circle
%.     x = 2cos(t), y = 2 sin(t), z=0
x_ := 2*cos(t)$  y_ := 2*sin(t)$  z_ := 0$
line_int := int(y_*df(x_,t) + z_*df(y_,t) + x_*df(z_,t), t,0,2*pi);

write "Stokes verified: ", (surface_int = line_int);
end;
```

-4π

Get *code*.



13.3.1 Exercises

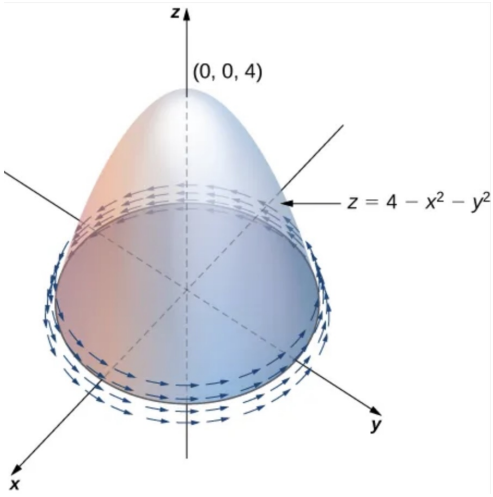
Exercise 92. (from [11, p.148, E13.1])
Use Stokes theorem to evaluate the line integral $\int_C -y^3dx + x^3dy + z^3dz$, where C is the intersection of the cylinder $x^2 + y^2 = 3$ and the plane $x + y + z = 1$ and the orientation is anticlockwise. $27\pi/2$
Hint: use the parameterization $\Phi(r, \theta) = (r\cos\theta, r\sin\theta)$, and show $\Phi_r \times \Phi_\theta = (0, 0, r)$

Exercise 93. (from [11, p.152, E13.4])
Verify Stokes theorem for the portion S of the surface $z = tan^{-1}(\frac{y}{x})$ which lies imside the cone $x^2 + y^2 + z^2$ and between the planes $z = 0$ and $z = 2\pi$ by using the vector field $F(x, y, z) = (xz, yz, -(x^2 + y^2))$. $-4\pi^3$



Do some selected examples and exercises from Stokes Theorem with REDUCE[excalc].
You should consult the book openSTAX for the mathematical considerations.

Exercise 94. Do some of the examples in *openSTAX*
a. Example 6.73



b. Example 6.74
c. Example 6.75

Exercise 95. Do checkpoint
a. Checkpoint 6.62
b. Checkpoint 6.35

Exercise 96. Do the following exercises
a. 335, 345, 367

13.4 Gauss's Theorem

Fact 20. (Theoretical minimum from WikiPediA: GDT)

A. *Classical Vector Calculus formulation.*
Let S be an oriented surface in $\mathbb{R}^3$, with outward normal $\mathbf{n}$, enclosing an open set U.
Let F be a smooth vector field defined on $\bar{U} := U \cup S$, i.e. $\partial U = S$.
Then we have[51] [52]

$$\iiint_U (div\,\mathbf{F})\; dx\, dy\, dz = \oiint_{\partial U} (\mathbf{F} \cdot \hat{\mathbf{n}})\, \mathrm{d}A \tag{13.8}$$

B. *Differential forms formulation.*

$$\int_U d\,\omega = \int_{\partial U} \omega \tag{13.9}$$

where $\omega \in \Omega^2\mathbb{R}^3$ is a 2-form on the smooth oriented volume U and ∂U is the closed surface (boundary) surrounding U with $\partial\omega \in \Omega^3\mathbb{R}^3$, a 3-form.

Mental image

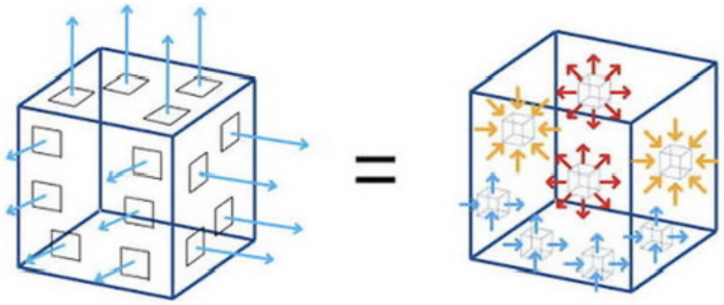


Figure 21: If $\mathbf{F}$ is the velocity of a gas then $div\,\mathbf{F}$ represents the rate of expansion (or compression) per unit volume. The divergence theorem states that the total expansion (or contraction) over a region U equals the total inflow (or outflow) across the boundary. [from Dineen, [11, p.178], picture from YouTube]

Mathematical reflection on the equivalence of (13.6) and (13.7)
We argue analog to the argumentation in §13.3. Let $F = (f_1, f_2, f_3) \equiv f_1 \cdot \vec{e_1} + f_2 \cdot \vec{e_2} + f_3 \cdot \vec{e_3}$ be the smooth vector field, where $\vec{e_1} = (1, 0, 0), ... \in \mathbb{R}^3$.

[51] We prefer the notation of the fact by Dineen [11, p.178]
[52] Motto: The integral of the normal component of a vector field over a closed surface equals the integral of its divergence over the enclosed volume. - From [31, p.924]

Translate F to its associated 1-form $\omega := f_1 \cdot dy \wedge dz + f_2 \cdot dz \wedge dx + f_3 \cdot dx \wedge dy \in \Omega^1\mathbb{R}^3$. Then we can write

$$\begin{aligned}
\int_{\partial U} \omega &\overset{p.d.}{=} \int_{\partial U} f_1 \cdot dy \wedge dz + f_2 \cdot dz \wedge dx + f_3 \cdot dx \wedge dy \overset{p.d.}{=} \iint_U F \bullet dU \\
&\overset{(13.7)}{=} \int_U d\omega = \int_U d(f_1 \cdot dy \wedge dz + f_2 \cdot dz \wedge dx + f_3 \cdot dx \wedge dy) \\
&= \int_U \left(\frac{\partial f_1}{\partial x} + \frac{\partial f_2}{\partial y} + \frac{\partial f_3}{\partial z} \right) \cdot dx \wedge dy \wedge dz \qquad \text{p.d. divergence} \\
&\ldots \qquad\qquad \text{using volume element } \; dV \; = \; dx\,dy\,dz \\
&= \iiint_U (\operatorname{div} \mathbf{F}) dV
\end{aligned}$$

□

Remark.
ad **A.** The left side is a volume integral over the volume U, and the right side is the surface integral over the boundary of the volume U. The closed set ∂U is oriented by outward-pointing normal, and $\hat{\mathbf{n}}$ is the outward pointing unit normal at almost each point on the boundary ∂U.
ad **B.** ω and $\partial\omega$ automatically respect the orientation.

Comment (About the notation $\nabla \cdot \mathbf{F}\, dV \equiv (div\, \mathbf{F})\, dx\, dy\, dz$).
1. We defined the *divergence* of the vector field $\mathbf{F} = (f_1, f_2, f_3) \equiv f_1\mathbf{i} + f_2\mathbf{j} + f_3\mathbf{k}$ as

$$div(\mathbf{F}) \overset{1}{:=} \nabla \cdot \mathbf{F} \overset{2}{:=} \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z} \right) \bullet (f_1, f_2, f_3) \tag{13.10}$$

$$\overset{3}{:=} \frac{\partial f_1}{\partial x} + \frac{\partial f_2}{\partial y} + \frac{\partial f_3}{\partial z} \tag{13.11}$$

1 : $\nabla \cdot \mathbf{F}$ is only meant as *symbolically* notation, because it's good to remember
2 : to denote the symbolically 'inner product' of $(\partial_x, \partial_y, \partial_z)$ with (f_1, f_2, f_3)
3 : both meaning the sum of the partial derivatives of $\mathbf{F}$'s components.

Only (13.11) is programmable.[53]

2. We show an implementation of *div* using package REDUCE[listvecops] as list procedure using formula (13.11), see example 50.2.

```
load_package listvecops;
listproc div(F);  df(F _1,x) + df(F _2,y) + df(F _3,z);
F := {-y^3,x^3,-z^3};
div(F);                  % = -3*z^2
```

Get *code*.

[53] 'Symbolism, however, does not prove anything ..', so the dictum by DINEEN in [11, p.177]

Example 52. Verify GAUSS divergence theorem for vector field $\vec{F} = (x - y, x + z, z - y)$ and surface S that consists of cone $x^2 + y^2 = z^2, 0 \leq z \leq 1$, and the circular top of the cone (see the figure below). Assume this surface S is oriented outward. – See Example 6.77.

Solution

The mathematical solution is given at openSTAX. We demonstrate here a solution with the REDUCE package `avector` for classical vector analysis and then with the package `excalc` for differential forms methods.

1st We calculate the LHS and RHS of (13.8) using REDUCE package `avector`.

```
load_package avector;

% Define the vector field
F := avec(x-y, x+z, z-y);

% 1. ---------- VOLUME Integral (LHS Divergence Side)

% Calculate divergence and triple integrate, radius = ra
Volume := defint(defint(defint( div(F) * ra, ra,0,z), z,0,1), th,0,2*pi );

% 2. ---------- SURFACE Integral (RHS)
%               we have to add the areas of a cone and his top

% 2.a ......... flux  through cone with parametrization by
R := avec( t*cos(th), t*sin(th), t);

% Calculate tangent vectors in direction t resp. th(eta)
Rt  := df(R,t);
Rth := df(R,th);

% Outward normal vector
N := Rth cross Rt;

% Substitute parametrization into F
Fc := avec(  t*cos(th)-t*sin(th),  t*cos(th)+t, t-t*sin(th) );

% Flux through the cone
flux_cone := defint( defint( Fc dot N, t,0,1), th,0,2*pi);

% 2.b .......... Flux through the circular top (z=1)

% Vector field Ft on z=1
Ft := avec(x-y,x+1,1-y);

% Substitute parametrization into F
```

```
Ft := avec( t*cos(th)-t*sin(th),  t*cos(th)+1, 1-t*sin(th) );

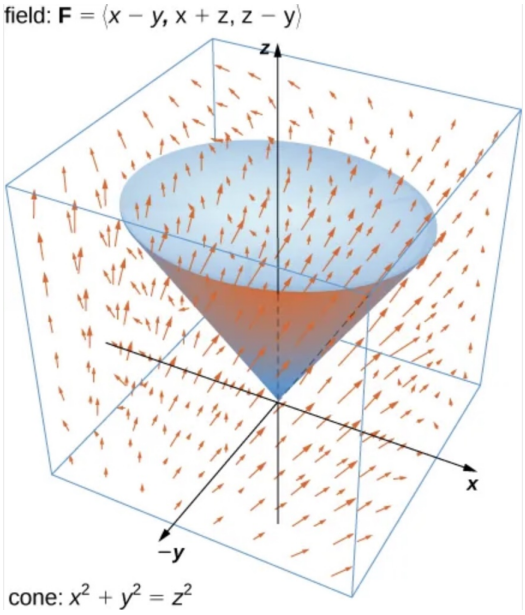
% Outward normal of top surface
Nt := avec(0,0,1);

% Use polar coordinates on the disk, radius = ra
flux_top := defint( defint( (Ft dot Nt) * ra, ra,0,1), th, 0,2*pi );

% check GAUSS theorem
total_flux := flux_cone + flux_top;
write "Check GAUSS theorem:", volume = total_flux;
```

$\frac{2\pi}{3} = \frac{2\pi}{3}$

Get *code*.



from OPENSTAX Example 6.77

2nd using exterior calculus of differential forms with REDUCE$^{\texttt{excalc}}$

```
load_package excalc;
pform {x,y,z}=0, omega=2;

% associated 2-form w.r.t. field F=(x-y,x+z,z-y)
omega := (x-y)*d y^d z + (x+z)*d z^d x + (z-y)*d x^d y;

% 1. ---------- VOLUME Integral (LHS Divergence Side)

% Compute the exterior derivative of omega and extract the divergence
div_F := (d omega) / (d x ^ d y ^ d z);

% Calculate the volume of cone using cylindrical coordinates
% with bounds 0 <= phi <= 2*pi, 0 <= z <= 1, 0 <= r <= z,  h-factor r

volume := int(int(int( div_F * r, r,0,z), z,0,1), phi, 0,2*pi);

% 2. ---------- SURFACE Integral (RHS)
%               we have to add the areas of a cone and his top
clear x,y,z;
pform {x,y,z, u,v}=0;

% 2.a .......... area cone
x := u*cos(v);   y := u*sin(v);   z := u;

% integrate with orientation sign -1
flux_cone := - int(int( coeffn(omega, d u^d v, 1), v, 0,2*pi), u, 0,1);

% 2.b .......... area top disk (z=1)
clear x,y,z;
pform {x,y,z, r,theta}=0;
omega := (x-y)*d y^d z + (x+z)*d z^d x + (z-y)*d x^d y;

% parametrization with cylindrical coordinates, 0<=r<=1, 0<=theta<=2 pi
x := r*cos(theta);   y := r*sin(theta);   z := 1;

flux_top  := int(int(coeffn(omega,d r^d theta,1),theta,0,2*pi),r,0,1);

total_flux := flux_cone + flux_top;
write "Check GAUSS theorem:", volume = total_flux;
```

$\frac{2\pi}{3} = \frac{2\pi}{3}$

Get *code*.

13.4.1 Exercises

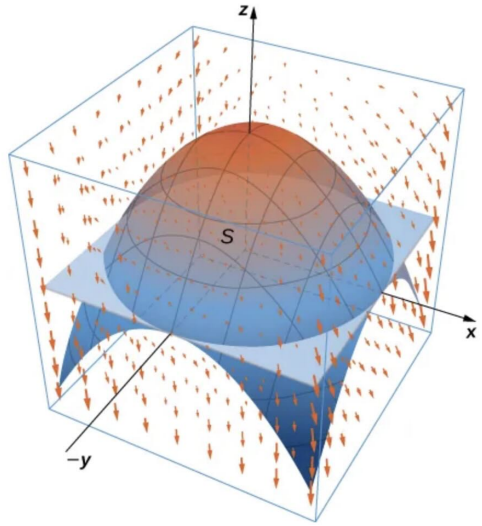
Exercise 97. (from [11, p.179, example 15.1])
Use the divergence theorem to evaluate the integral $\iiint_V div(F)dV$, where V is the solide cylinder $\{(x, y, z) : x^2 + y^2 < 1, 0 < z < 1\}$ and the vector field is
$F(x, y, z) = (1 - (x^2 + y^2)^3, 1 - (x^2 + y^2)^3, x^2z^2)$. $\pi/4$

Exercise 98. (from [11, p.188, exercise 15.5])
Use the divergence theorem to prove $\iint_S x^2, -2xy, 3xz) = 3\pi$, where S is the outwardly oriented boundary of the region $\{(x, y, z) : x \geq 0, y \geq 0, z \geq 0, x^2 + y^2 + z^2 \leq 4\}$.



Do some selected examples and exercises from Gauss with REDUCE[excalc].
You should consult the book openSTAX for the mathematical considerations.

Exercise 99. Do some of the examples in *openSTAX*:



openSTAX exercise 410

a. Example 6.78
b. Example 6.79

Exercise 100. Do some of the checkpoints
a. Checkpoint 6.66
b. Checkpoint 6.67

Remark. ARNOLD [3, p.194] gives the following summary of the generalized Stokes:

$$\begin{aligned} f(y) - f(x) &= \int_\ell \mathbf{grad} f d\mathbf{l} && \text{if } \partial\ell = y - x \\ \int_\ell \mathbf{A}\, d\mathbf{l} &= \iint_S \mathbf{curl}\, \mathbf{A} \cdot d\mathbf{n} && \text{if } \partial S = \ell \\ \iint_S \mathbf{A}\, d\mathbf{n} &= \iiint_D (div\ \mathbf{A})\omega^3 && \text{if } \partial D = S \end{aligned}$$

where ω^3 is the volume element in $D \subset \mathbb{R}^3$

14 Bibliography

References

[1] Abraham, R. & J. E. Marsden, D. (1987): *Foundations of Mechanics, Second Edition.* Addison-Wesley Publishing Company, Inc.
url: https://authors.library.caltech.edu/records/2y9mc-bgv54/files/FoM2.pdf

[2] Arapura, D. (2016): *Introduction to Differential Forms.*
url: https://www.math.purdue.edu/~dvb/preprints/diffforms.pdf

[3] Arnold, V. I. (1978): *Mathematical Methods of Classical Mechanics.*
New York: Springer.

[4] Bachman, D. (2nd., 2012): *A Geometric Approach to Differential Forms.*
Boston: Birkhäuser.

[5] Banchoff, T. F. & Lovett, S. (^{3}2022): *Differential Geometry of Curves and Surfaces.* London: Chapman and Hall.

[6] Bishop, R.L. & Goldberg, S.I. (1980): *A Tensor Analysis on Manifolds.*
New York: Dover Publ.

[7] Brehmer, S. & Haar, H. (1973): *Differentialformen und Vektoranalysis.*
Berlin: VEB Deutscher Verlag der Wissenschaften.

[8] Bryan, K. (?): *Differential Forms.*
url: https://www.rose-hulman.edu/~bryan/lottamath/difform.pdf

[9] Collier, B. (2022): *A Beginner's Guide to Differential forms.*
ISBN 9780957389472: Incomprehensible Books.

[10] do Carmo, M.P. (1992): *Differential Forms and Applications.*
Brasil: preliminary lecture notes.

[11] Dineen, S. (^{2nd}2001): *Multivariate Calculus and Geometry.*
London: Springer.

[12] Eliashberg, Y. (2012): *Math 52H: Multilinear algebra, differential forms and Stokes' theorem.* url: http://math.stanford.edu/~eliash/Public/52h-2012/52htext.pdf

[13] Hearn, A.C.& Schöpf, R. et al. (2026): *REDUCE User's Manual. Free Version.*
url: https://reduce-algebra.sourceforge.io/manual/manual.pdf
url: https://reduce-algebra.sourceforge.io/manual/manual.html

[14] Holmann, H. & Rummler, H. (1972): *Alternierende Differentialformen.*
Mannheim: B.I. Wissenschaftsverlag.

[15] Hubbard, J. H. & Hubbard, B. B. (1999): *Vector Calculus, Linear Algebra, and Differential Forms.* New York: Prentice Hall Inc.

[16] Jänich, K. (1992): *Vektoranalysis.*
Berlin: Springer.

[17] Krysicki, W. & Włodaski, L. (1971): *Höhere Mathematik in Aufgaben. - Teil 2.*
Leipzig: BSB B.G. Teubner.

[18] Lebl, J. (2025): *Differential Forms Crash Course.*
Berlin: Springer.

[19] Lindner, W. (2026): *Some Tapas of Tensors.* (using Eigenmath)
url: `https://lindnerdrwg.github.io/Some-Tapas-of-Tensors.pdf`

[20] Lindner, W. (2026): *Tensor Calculus with Maxima-Online.*
url: `https://lindnerdrwg.github.io/TensorCalculus.pdf`

[21] Lindner, W. (2026): *Classical Differential Geometry of Surfaces with Maxima-Online.* url: `https://lindnerdrwg.github.io/DiffGeo-of-Surfaces-Maxima.pdf`

[22] Lindner, W. (2026): *Differential Geometry of Surfaces using Eigenmath.*
url: `https://lindnerdrwg.github.io/Differential-Geometry-of-Surfaces.pdf`

[23] Lindner, W. (2021): *Linear Algebra Interactive! Part 3: Analytical-Geometry*
url: `https://lindnerdrwg.github.io/LAi3-Singular-Systems-Analytical-Geometry`

[24] Lindner, W. (2021): *Linear Algebra Interactive! Part 5: Determinants*
url: `https://lindnerdrwg.github.io/LAi5-Determinants.pdf`

[25] Lindner, W. (2011): *APOS Theory and A.C.E. Teaching Cycle.*
url: `https://lindnerdrwg.github.io/CASinEDU-2-APOS.pdf`

[26] Lindner, W. (2011): *Concept formation according to APOS theory.*
url: `https://lindnerdrwg.github.io/CASinEDU-2B.pdf`

[27] Lipschutz, M. M. (1969): *Differential Geometry.* New York: McGraw-Hill

[28] Long, N. & Keller, M.T. & Keller (2011): *Active Calculus - Multivariable* 2^{nd}.
url: `https://activecalculus.org/multi2e/root-1-2.html`

[29] Lovelock, D. & Rund, H. (1975): *Tensors, Differential Forms, and Variational Principles.* New York: Dover Publications.

[30] Melton, R. (2026): *Lumen Learning : Calculus III.*
url: `https://courses.lumenlearning.com/calculus3/`

[31] Marsden, J. & Weinstein, A. ([2]1985): *Calculus I, II, III.* New York: Springer.
url: https://authors.library.caltech.edu/records/hb74n-f9x09/files/Calc2w.pdf
url: https://authors.library.caltech.edu/records/rg25s-crx26/files/Calc3w.pdf

[32] Mazurowski, L. (2011): *Differential Forms Lecture Notes.*
url: https://e.math.cornell.edu/people/mazurowski/DifferentialFormsNotes.pdf

[33] Minorski, W. P. (1965): *Aufgabensammlung der höheren Mathematik. 2. Auflage* Leipzig: VEB Fachbuchverlag.

[34] openSTAX (2026): *Multiple Integration.*
url: https://openstax.org/books/calculus-volume-3/pages/5-introduction

[35] openSTAX (2026): *Vector Calculus Integration.*
url: https://openstax.org/books/calculus-volume-3/pages/6-introduction

[36] O'Neill, B. (1966): *Elementary Differential Geometry.*
New York: Academic Press.

[37] Needham, T. (2021): *Visual Differential Geometry and Forms.*
Princeton: Princeton University Press.

[38] Reinhardt, F. & Soeder, H. (1977): *dtv-Atlas zur Mathematik -Tafeln und Texte Bd. 2.* München: Deutscher Taschenbuch Verlag.

[39] Riebesehl, D. (2024): *Vektoranalysis und Differentialformen.*
Heidelberg: Springer.

[40] Roolfs, G. (2026): *Vektorfeld. Polarkoordinaten.*
url: http://groolfs.de/Verschiedenespdf/VektorfeldPolarkoordinaten.pdf

[41] Schey, H.M. (4th ed. 2005): *div, grad, curl and all that.*
New York: W.W. Norton.

[42] Schrüfer, E. (2004): *EXCALC: A System for Doing Calculations in the Calculus of Modern Differential Geometry.* Sankt Augustin: GMD Schloss Birlinghoven.
url: https://reduce-algebra.sourceforge.io/manual/manualse147.html#x170-82600020.21

[43] Schulz, W. & Schulz, A. (2012): *A Practical Introduction to Differential Forms.*
url: https://web.mit.edu/al24406/www/mathmeth/DiffForms_SchulzSchulz_10Sep.pdf

[44] Shuvo, S.J. (?): *Introduction to Differential forms.* url: ?.pdf

[45] Singee, I.M. & Thorpe, J.A. (2001): *Lecture Notes on Elementary Topology and Geometry* New York: Springer.

[46] Sjamaar, R. (2017): *Manifolds and Differential Forms.*
url: https://pi.math.cornell.edu/~sjamaar/manifolds/manifold.pdf

[47] Steeb, W.-H. & Lewien, D. (1992): *Algorithms and Computation with* Reduce. Mannheim: Bibliographisches Institut.

[48] Thorpe, J. A. (1979): *Elementary Topics in Differential Geometry.* New York: Springer.

[49] Weigt, G. (2025): *Eigenmath Manual.* url: `https://georgeweigt.github.io/eigenmath.pdf`

[50] Wright, F.J. (2018): Reduce *Calculus Tutorial.* url: `https://reduce-algebra.sourceforge.io/tutorials/calculus-tutorial.html`

[51] Wright, F.J. (2018): Reduce *Vectors and Linear Algebra Tutorial.* url: `https://reduce-algebra.sourceforge.io/tutorials/veclinalg-tutorial.html`

⋈

Dr. Wolfgang G. Lindner
Leichlingen, Germany
`dr.w.g.Lindner@gmail.com`
2027