



INTERNATIONAL ATOMIC ENERGY AGENCY  
UNITED NATIONS EDUCATIONAL, SCIENTIFIC AND CULTURAL ORGANIZATION



**INTERNATIONAL CENTRE FOR THEORETICAL PHYSICS**

34100 TRIESTE (ITALY) • P.O.B. 586 • MIRAMARE • STRADA COSTIERA 11 • TELEPHONE: 8840-1  
CABLE: CENTRATOM • TELEX 460392 - I

SECOND SCHOOL ON ADVANCED TECHNIQUES  
IN COMPUTATIONAL PHYSICS  
(18 January - 12 February 1988)

SMR.282/ 18

USER'S MANUAL

Eberhard Schrufer  
GMD, St. Augustin, West Germany

EXCALC

A System for Doing Calculations in the Calculus  
of  
Modern Differential Geometry

USER's MANUAL

Eberhard Schröder

April 1986

Acknowledgments

This program was developed over several years. I would like to express my deep gratitude to Dr. Anthony Hearn for his continuous interest in this work, and especially for his hospitality and support during a visit in 1984/85 at the Rand Corporation, where substantial progress on this package could be achieved. The Heinrich Hertz-Stiftung supported this visit. Many thanks are also due to Drs. F.W. Hehl, University of Cologne, and J.D. McCrea, University College Dublin, for their suggestions and work on testing this program.

Table of Contents

Introduction

1. Declarations
2. Exterior Multiplication
3. Partial Differentiation
4. Exterior Differentiation
5. Inner Product
6. Lie Derivative
7. Hodge-\* Duality Operator
8. Variational Derivative
9. Handling of Indices
10. Metric Structures
11. Riemannian Connections
12. Ordering and Structuring
13. Appendix
  - A1. Summary of Operators and Commands
  - A2. Examples

INTRODUCTION

EXCALC is designed for easy use by all who are familiar with the calculus of Modern Differential Geometry. Its syntax is kept as close as possible to standard textbook notations. Therefore, no great experience in writing computer algebra programs is required. It is almost possible to input to the computer the same as what would have been written down for a handcalculation. For example, the statement

$$f^*x^*y + u_1(y^*z^*x)$$

would be recognized by the program as a formula involving exterior products and an inner product. The program is currently able to handle scalar-valued exterior forms, vectors and operations between them, as well as non-scalar valued forms (indexed forms). With this, it should be an ideal tool for studying differential equations, doing calculations in general relativity and field theories, or doing simple things as calculating the Laplacian of a tensor field for an arbitrary given frame. With the increasing popularity of this calculus, this program should have an application in almost any field of physics and mathematics.

Since the program is completely embedded in REDUCE, all features and facilities of REDUCE are available in a calculation. Even for those who are not quite comfortable in this calculus, there is a good chance of learning it by just playing with the program.

This is still a very experimental version, and changes of the syntax are to be expected. The performance of the program can still be increased considerably.

Complaints and comments are appreciated and should be sent to:

Dr. Eberhard Schröder  
Institut fuer methodische Grundlagen (Fl-P)  
Gesellschaft fuer Mathematik und Datenverarbeitung mbH  
Postfach 1240  
Schloss Birlinghoven  
D-5205 St. Augustin 1  
WEST GERMANY

If the use of this program leads to a publication, this document should be cited, and a copy of the article should be sent to the above address.

## 1. Declarations

Geometrical objects like exterior forms or vectors are introduced to the system by declaration commands. The declarations can appear anywhere in a program, but must, of course, be made prior to the use of the object. Everything that has no declaration is treated as a constant; therefore zero-forms must also be declared.

An exterior form is introduced by

```
PFORM declaration1, declaration2, ...;
```

where

```
declaration::=<name>=<number>|<identifier>|<expression>
<name>::=<identifier>|<identifier>(<arguments>)
```

For example

```
PFORM U=K, V=4, F=0, W=DIM-1;
```

would declare U to be an exterior form of degree K, V to be a form of degree 4, F to be a form of degree 0 (a function), and W to be a form of degree DIM-1.

If the exterior form should have indices, the declaration would be

```
PFORM CURV(A,B)=2, CHRIS(A,B)=1;
```

The name of the indices is arbitrary.

The declaration of vectors is similar. The command TVECTOR takes a list of names.

Example:

To declare X as a vector and COMM as a vector with two indices, one would say

```
TVECTOR X, COMM(A,B);
```

If a declaration of an already existing name is made, the old declaration is removed, and the new one is taken.

## 2. Exterior Multiplication

Exterior multiplication between exterior forms is carried out with the nary infix operator ^ (wedge). Factors are ordered according to the usual ordering in REDUCE using the commutation rule for exterior products.

Examples:

```
PFORM U=1, V=1, W=K;
```

```
U^V;
```

```
U^V
```

```
V^U;
```

```
- U^V
```

```
U^U;
```

```
0
```

```
W^U^V;
```

```
      K
(- 1) *U^V^W
```

```
(3*U-A*W)^(W+5*V)^U;
```

```
A*(5*U^V^W - U^W^W)
```

It is possible to declare the dimension of the underlying space by

```
SPACEDIM <number>|<identifier>;
```

If an exterior product has a degree higher than the dimension of the space, it is replaced by 0:

```
SPACEDIM 4;
```

```
PFORM U=2, V=3;
```

```
U^V;
```

```
0
```

## 3. Partial Differentiation

Partial differentiation is denoted by the operator @. Its capability is the same as the REDUCE DF operator.

Examples:

```

@ (SIN X,X);
COS(X)
@ (F,X);
0

```

An identifier can be declared to be a function of certain variables. This is done with the command FDOMAIN. The following would tell the partial differentiation operator that F is a function of the variables X and Y and that H is a function of X.

```
FDOMAIN F=F(X,Y),H=H(X);
```

Applying @ to F and H would result in

```

@ (F,X);
@ F
X
@ (X*F,X);
F + X*@ F
X
@ (H,Y);
0

```

The partial derivative symbol can also be an operator with a single argument. It then represents a natural base element of a tangent vector.

Example:

```

A*@ X + B*@ Y;
A*@ + B*@
X Y

```

#### 4. Exterior Differentiation

Exterior differentiation of exterior forms is carried out by the operator d. Products are normally differentiated out, i.e.

```

PFORM X=0,Y=K,Z=M;
D(X * Y);
X*d Y + d X*Y
D(R*Y);
R*d Y
D(X*Y^Z);
K
(- 1) *X*Y^d Z + X*d Y^Z + d X*Y^Z

```

This expansion can be suppressed by the command NOXPND D.

```

NOXPND D;
D(Y^Z);
d(Y^Z)

```

To obtain a canonical form for an exterior product when the expansion is switched off, the operator D is shifted to the right if it appears in the leftmost place.

```

D Y ^ Z;
K
- (- 1) *Y^d Z + d(Y^Z)

```

Expansion is performed again when the command XPND D is executed.

Functions which are implicitly defined by the FDOMAIN command are expanded into partial derivatives:

```

PFORM X=0,Y=0,Z=0,F=0;
FDOMAIN F=F(X,Y);
D F;
@ F*d X + @ F*d Y
X Y

```

If an argument of an implicitly defined function has further dependencies the chain rule will be applied e.g.

FDOMAIN Y=Y(Z);

D F;

$\frac{\partial}{\partial X} F \frac{dX}{dZ} + \frac{\partial}{\partial Y} F \frac{dY}{dZ} + \frac{\partial}{\partial Z} F \frac{dZ}{dZ}$

Expansion into partial derivatives can be inhibited by NOXPND @ and enabled again by XPND @.

The operator is of course aware of the rules that a repeated application always leads to zero and that there is no exterior form of higher degree than the dimension of the space.

D D X;

0

PFORM U=K;

SPACEDIM K;

D U;

0

### 5. Inner Product

The inner product between a vector and an exterior form is represented by the diphthong | (underscore or-bar), which is the notation of many textbooks. If the exterior form is an exterior product, the inner product is carried through any factor

Example:

PFORM X=0,Y=K,Z=M;

TVECTOR U,V;

U| (X\*Y\*Z);

$X*((-1)^K * Y^* U_{|Z} + U_{|Y}^* Z)$

In repeated applications of the inner product to the same exterior form the vector arguments are ordered e.g.

(U+X\*V)| (U<sub>|</sub> (3\*Z));

- 3\*U<sub>|</sub>V<sub>|</sub>Z

The duality of natural base elements is also known by the system, i.e.

PFORM X=0,Y=0;

(A\*U X+B\*U (Y))| (3\*D X-D Y);

3\*A - B

### 6. Lie Derivative

The Lie derivative can be taken between a vector and an exterior form or between two vectors. It is represented by the infix operator |. In the case of Lie differentiating, an exterior form by a vector, the Lie derivative is expressed through inner products and exterior differentiations, i. e.

PFORM Z=K;

TVECTOR U;

U | Z;

U<sub>|</sub>d Z + d(U<sub>|</sub>Z)

If the arguments of the Lie derivative are vectors, the vectors are ordered using the anticommutivity property, and functions (zero forms) are differentiated out.

Example:

TVECTOR U,V;

V | U;

- U<sub>|</sub>V

PFORM X=0,Y=0;

(X\*U)| (Y\*V);

- U\*Y\*V<sub>|</sub>d X + V\*X\*U<sub>|</sub>d Y + X\*Y\*U<sub>|</sub>V

### 7. Hodge-\* Duality Operator

The Hodge-\* duality operator maps an exterior form of degree K to an exterior form of degree N-K, where N is the dimension of the space. The double application of the operator must lead back to the original exterior form up to a factor. The following example shows how the factor is chosen here

```
SPACEDIM N;
PFORM X=K;
# # X;
      2
      (K + K*N)
(- 1) *X*SGN
```

The indeterminate SGN in the above example denotes the sign of the determinant of the metric. It can be assigned a value or will be automatically set if more of the metric structure is specified (via COFRAME), i.e. it is then set to  $g/|g|$ , where  $g$  is the determinant of the metric. If the Hodge-\* operator appears in an exterior product of maximal degree as the leftmost factor, the Hodge-\* is shifted to the right according to

```
PFORM X=K,Y=K;
# X ^ Y;
      2
      (K + K*N)
(- 1) *X^# Y
```

More simplifications are performed if a coframe is defined.

### 8. Variational Derivative

The function VARDF returns as its value the variation of a given Lagrangian n-form with respect to a specified exterior form (a field of the Lagrangian). In the shared variable BNDEQ!\*, the expression is stored that has to yield zero if integrated over the boundary.

Syntax:

```
VARDF(<Lagrangian n-form>,<exterior form>)
```

Example:

```
SPACEDIM 4;
PFORM L=4,A=1,J=3;
L:=-1/2*D A ^ # D A - A^# J$ %Lagrangian of the e.m. field
VARDF(L,A);
- (# J + d # d A) %Maxwell's equations
BNDEQ!*;
- 'A^# d A %Equation at the boundary
```

Restrictions:

In the current implementation, the Lagrangian must be built up by the fields and the operations d, #, and @. Variation with respect to indexed quantities is currently not allowed.

For the calculation of the conserved currents induced by symmetry operators (vector fields), the function NOETHER is provided. It has the syntax:

```
NOETHER(<Lagrangian n-form>,<field>,<symmetry generator>)
```

Example:

```
pform l=4,a=1,f=2;
spacedim 4;
l:= -1/2*d a^#d a; %Free Maxwell field;
tvector x(k); %An unspecified generator;
noether(l,a,x(-k));
(- 2*d(X_#A)^# d A - (X_#d A)^# d A + d A^#(X_#d A))/2
      K      K      K
```

Comment: The above expression would be the canonical energy-momentum 3-forms of the Maxwell field, if X is interpreted as a translation;

### 9. Handling of Indices

Exterior forms and vectors may have indices. On input, the indices are given as arguments of the object. A positive argument denotes a superscript and a negative argument a subscript. On output, the indexed quantity is displayed two dimensionally if NAT is on. Indices may be identifiers or numbers. However, zero is currently not allowed to be an index.

Example:

PFORM OM(K,L)=M,E(K)=1;

E(K)^E(-L);

K  
E ^E  
L

OM(4,-2);

4  
OM  
2

In the current release, full simplification is performed only if an index range is specified. It is hoped that this restriction can be removed soon. If the index range (the values that the indices can obtain) is specified, the given expression is evaluated for all possible index values, and the summation convention is understood.

Example:

INDEXRANGE T,R,PH,Z;

PFORM E(K)=1,S(K,L)=2;

W := E(K)\*E(-K);

W := E<sup>T</sup><sub>T</sub> \* E<sup>R</sup><sub>R</sub> + E<sup>PH</sup><sub>PH</sub> \* E<sup>Z</sup><sub>Z</sub>

S(K,L):=E(K)^E(L);

T T  
S := 0

R T T R  
S := - E ^E

PH T T PH  
S := - E ^E

If the expression to be evaluated is not an assignment, the values of the expression are displayed as an assignment to an indexed variable

with name NS. This is done only on output, i.e. no actual binding to the variable NS occurs.

E(K)^E(L);

T T  
NS := 0

R T T R  
NS := - E ^E

It should be noted, however, that the index positions on the variable NS can sometimes not be uniquely determined by the system (because of possible reorderings in the expression). Generally it is advisable to use assignments to display complicated expressions.

In certain cases, one would like to inhibit the summation over specified index names, or at all. For this the command NOSUM <indexname>,...; and the switch NOSUM are available. The command NOSUM has the effect that summation is not performed over those indices which had been listed. The command RENOSUM enables summation again. The switch NOSUM, if on, inhibits any summation.

It is possible to declare an indexed quantity completely antisymmetric or completely symmetric by the command ANTISYMMETRIC <name>,...; or SYMMETRIC <name>,...;. If applicable, these commands should be issued, since great savings in memory and execution time result. Only strict components are printed.

## 10. Metric Structures

A metric structure is defined in EXCALC by specifying a set of basis one-forms (the coframe) together with the metric.

Syntax:

COFRAME <identifier><(index1)>=<expression1>,  
<identifier><(index2)>=<expression2>,  
.  
.  
<identifier><(indexn)>=<expressionn>  
WITH METRIC <name>=<expression>;

This statement automatically sets the dimension of the space and the index range. The clause WITH METRIC can be omitted if the metric is Euclidean and the shorthand WITH SIGNATURE <diagonal elements> can be used in the case of a pseudo-Euclidean metric. The splitting of a metric structure in its metric tensor coefficients and basis one-forms is

completely arbitrary including the extrema of an orthonormal frame and a coordinate frame.

Examples:

```
COFRAME E R=D R, E(PH)=R*D PH
  WITH METRIC G=E(R)*E(R)+E(PH)*E(PH);      %Polar coframe;

COFRAME E(R)=D R, E(PH)=R*D(PH);            %Same as before;

COFRAME O(T)=D T, O X=D X
  WITH SIGNATURE -1,1;                        %A Lorentz coframe;

COFRAME B(XI)=D XI, B(ETA)=D ETA
  WITH METRIC W=-1/2*(B(XI)*B(ETA)+B(ETA)*B(XI)); %A lightcone coframe;

COFRAME E R=D R, E PH=D PH
  WITH METRIC G=E R+E R+R**2*E PH*E PH;      %Polar coordinate
                                              %basis;
```

Individual elements of the metric can be accessed just by calling them with the desired indices. The value of the determinant of the covariant metric is stored in the variable DETM!\*. The metric is not needed for lowering or raising of indices as the system performs this automatically, i.e. no matter in what index position values were assigned to an indexed quantity, the values can be retrieved for any index position just by writing the indexed quantity with the desired indices.

Example:

```
COFRAME E T=D T, E X=D X, E Y=D Y
  WITH SIGNATURE -1,1,1;

PFORM F(K,L)=0;

ANTISYMMETRIC F;

F(-T,-X):=EX$ F(-X,-Y):=B$ F(-T,-Y):=0$
ON NERO;

F(K,-L):=F(K,-L);

X
F  := - EX
T

T
F  := - EX
X

Y
F  := - B
X

X
F  := B
Y
```

Any expression containing differentials of the coordinate functions will be transformed into an expression of the basis one-forms. The system also knows how to take the exterior derivative of the basis one-forms.

Example (spherical coordinates):

```
COFRAME E(R)=D(R), E(TH)=R*D(TH), E(PH)=R*SIN(TH)*D(PH);

D R^D TH;

R TH
(E ^E )/R

D(E(TH));

R TH
(E ^E )/R
```



```

PFORM F=0;

FDOMAIN F=F(R,TH,PH);

FACTOR E;

ON RAT;

D F;          %The "gradient" of F in spherical coordinates;

      R          TH          PH
E *% F + (E *% F)/R + (E *% F)/(R*SIN(TH))
      R          TH          PH

```

The frame dual to the frame defined by the COFRAME command can be introduced by FRAME <identifier>. This command causes the dual property to be recognized, and the tangent vectors of the coordinate functions are replaced by the frame basis vectors.

Example:

```

COFRAME B R=D R,B PH=R*D PH,E Z=D Z;    %Cylindrical coframe;

FRAME X;

ON NERO;

X(-K)_|B(L);

      R
NS   := 1
      R

      PH
NS   := 1
      PH

      Z
NS   := 1
      Z

X(-K) |_ X(-L);          %The commutator of the dual frame;

NS   := X /R
      PH R      PH

NS   := ( - X )/R        %I. e. it is not a coordinate base;
      R PH      PH

```

As a convenience, the frames can be displayed at any point in a program by the command DISPLAYFRAME;

The Hodge-\* duality operator returns the explicitly constructed dual element if applied to coframe base elements. The metric is properly taken into account.

The total antisymmetric Levi-Cevita tensor EPS is also available. The value of EPS with an even permutation of the indices in a covariant position is taken to be +1.

## 11. Riemannian Connections

The command RIEMANNCONX is provided for calculating the connection 1 forms. The values are stored on the name given to RIEMANNCONX. This command is far more efficient than calculating the connection from the differential of the basis one-forms and using inner products.

Example (Calculate the connection 1-form and curvature 2-form on S(2)):

```

COFRAME E TH=R*D TH,E PH=R*SIN(TH)*D PH;

RIEMANNCONX OM;

OM(K,-L);          %Display the connection forms;

      TH
NS   := 0
      TH

      PH          PH
NS   := (E *COS(TH))/(SIN(TH)*R)
      TH

      TH          PH
NS   := ( - E *COS(TH))/(SIN(TH)*R)
      PH

      PH
NS   := 0
      PH

```

```
PFORM CURV(K,L)=2;

CURV(K,-L):=D OM(K,-L) + OM(K,-M)^OM(M-L);  %The curvature forms

TH
CURV      := 0
TH

PH      TH PH 2
CURV      := (- E ^E )/R      %Of course it was a sphere with
TH                                     %radius R.

TH      TH PH 2
CURV      := (E ^E )/R
PH

PH
CURV      := 0
PH
```

## 12. Ordering and Structuring

The ordering of an exterior form or vector can be changed by the command FORDER. In an expression, the first identifier or kernel in the arguments of FORDER is ordered ahead of the second, and so on, and ordered ahead of all not appearing as arguments. This ordering is done on the internal level and not only on output. The execution of this statement can therefore have tremendous effects on computation time and memory requirements. REMFORDER brings back standard ordering for those elements that are listed as arguments.

Another ordering command is ISOLATE. It takes one argument. The system attempts to shift out this identifier or kernel to the leftmost position, utilizing commutation and derivative rules. REMISOLATE restores normal ordering.

Example:

```
PFORM U=K,V=L,W=M;

U^D(V)^W;

U^d V^W

FORDER V;

U^D(V)^W;

(K*L + K)
(- 1)      *d V^U^W
```

```
ISOLATE V;

U^D(V);

(K*L + K)      L
(- 1)      *d(V^U) - (- 1) *V^d U
```

An expression can be put in a more structured form by renaming a subexpression. This is done with the command KEEP which has the syntax

```
KEEP <name1>=<expression1>,<name2>=<expression2>,...
```

The effect is that rules are set up for simplifying <name> without introducing its definition in an expression. In an expression the system also tries by reordering to generate as many instances of <name> as possible.

Example:

```
PFORM X=0,Y=0,Z=0,F=0,J=3;

KEEP J=D X^D Y^D Z;

J;

J

D J;

0

J^D X;

0

FDOMAIN F=F(X);

D F^D Y^D Z;

0 F^J
X
```

The capabilities of KEEP are currently very limited. Only exterior products should occur as righthand sides in KEEP.

### 13. Appendix

#### A1. Summary of Operators and Commands

^	Exterior Multiplication	2
@	Partial Differentiation	3
@	Tangent Vector	3
#	Hodge-* Operator	7
_	Inner Product	5
_	Lie Derivative	6
ANTISYMMETRIC	Declares completely antisymmetric indexed quantities	9
COFRAME	Declaration of a coframe	10
d	Exterior differentiation	2
DISPLAYFRAME	Displays the frame	10
EPS	Levi-Civita tensor	10
FDOMAIN	Declaration of implicit dependencies	3
FORDER	Ordering command	12
FRAME	Declares the frame dual to the coframe	10
INDEXRANGE	Declaration of indices	9
ISOLATE	Ordering command	12
KEEP	Structuring command	12
METRIC	Clause of COFRAME to specify a metric	10
NOETHER	Calculates the Noether current	8
NOSUM	Inhibits summation convention	9
NOXPND d	Inhibits the use of product rule for d	4
NOXPND @	Inhibits expansion into partial derivatives	4
PFORM	Declaration of exterior forms	1
REMFORDER	Clears ordering	12
REMISOLATE	Clears ISOLATE command	12
RENOSUM	Enables summation convention	9
RIEMANNCONX	Calculation of a Riemannian Connection	11
SIGNATURE	Clause of COFRAME to specify a pseudo-Euclidean metric	10
SPACEDIM	Command to set the dimension of a space	2
SYMMETRIC	Declares completely symmetric indexed quantities	9
TVECTOR	Declaration of vectors	1
VARDF	Variational derivative	8
XPND d	Enables the use of product rule for d (is set on default)	4
XPND @	Enables expansion into partial derivatives (is set on default)	4

#### A2. Examples

The following examples should illustrate the use of EXCALC. It is not intended to show the most efficient or most elegant way of stating the problems; rather the variety of syntactic constructs are exemplified. The examples are on a test file distributed with EXCALC.

```
%Problem: Calculate the PDE's for the isovector of the heat equation.
%-----
%          (c.f. B.K. Harrison, f.B. Estabrook, "Geometric Approach...",
%          J. Math. Phys. 12, 653, 1971);

%The heat equation @ psi = @ psi is equivalent to the set of exterior
%          xx          t

%equations (with u=@ psi, y=@ psi):
%          T          x

pform psi=0,u=0,x=0,y=0,t=0,a=1,da=2,b=2;

a:=d psi - u*d t - y*d x;

da:=- d u*d t - d y*d x;

b:=u*d x*d t - d y*d t;

%Now calculate the PDE's for the isovector;

tvector v;

pform vpsi=0,vt=0,vu=0,vx=0,vy=0;
fdomain vpsi=vpsi(psi,t,u,x,y),vt=vt(psi,t,u,x,y),vu=vu(psi,t,u,x,y),
vx=vx(psi,t,u,x,y),vy=vy(psi,t,u,x,y);

v:=vpsi*@ psi + vt*@ t + vu*@ u + vx*@ x + vy*@ y;

factor d;
on rat;

i1:=v |_ a - l*a;

pform o=1;

o:=ot*d t + ox*d x + ou*d u + oy*d y;

fdomain f=f(psi,t,u,x,y);

i11:=v |_d a - l*a + d f;

let vx=-@ (f,y),vt=-@ (f,u),vu=@ (f,t)+u*@ (f,psi),vy=@ (f,x)+y*@ (f,psi),
vpsi=f-u*@ (f,u)-y*@ (f,y);

factor ^;

i2:=v |_ b - x1*b - o^a + zet*da;
```

```

let ou=0,oy=@(f,u,psi),ox=-u*@ (f,u,psi),
    ot=@(f,x,psi)+u*@ (f,y,psi)+y*@ (f,psi,psi);
12;

let zet=-@ (f,u,x)-@ (f,u,y)*u-@ (f,u,psi)*y;
12;

let xi=-@ (f,t,u)-u*@ (f,u,psi)+@ (f,x,y)+u*@ (f,y,y)+y*@ (f,y,psi)+@ (f,psi);
12;

let @ (f,u,u)=0;
12;      % These PDE's have to be solved.

```

```

clear a,da,b,v,i1,i11,o,i2,xi;
remfdomain f;
clear @ (f,u,u);

```

%Problem:

```

%-----
%Calculate the integrability conditions for the system of PDE's:
%(c.f. B.F. Schutz, "Geometrical Methods of Mathematical Physics"
%Cambridge University Press, 1984, p. 156)

```

```

% @ z / @ x + a1*z + b1*z = c1
% 1 1 2

```

```

% @ z / @ y + a2*z + b2*z = c2
% 1 1 2

```

```

% @ z / @ x + f1*z + g1*z = h1
% 2 1 2

```

```

% @ z / @ y + f2*z + g2*z = h2
% 2 1 2

```

```

pform w(k)=1,integ(k)=4,z(k)=0,x=0,y=0,a=1,b=1,c=1,f=1,g=1,h=1,
    a1=0,a2=0,b1=0,b2=0,c1=0,c2=0,f1=0,f2=0,g1=0,g2=0,h1=0,h2=0;

```

```

fdomain a1=a1(x,y),a2=a2(x,y),b1=b1(x,y),b2=b2(x,y),
    c1=c1(x,y),c2=c2(x,y),f1=f1(x,y),f2=f2(x,y),
    g1=g1(x,y),g2=g2(x,y),h1=h1(x,y),h2=h2(x,y);

```

```

a:=a1*d x+a2*d y$

```

```

b:=b1*d x+b2*d y$
c:=c1*d x+c2*d y$
f:=f1*d x+f2*d y$
g:=g1*d x+g2*d y$
h:=h1*d x+h2*d y$

```

%The equivalent exterior system:

```

factor d;
w(1) := d z(-1) + z(-1)*a + z(-2)*b - c;
w(2) := d z(-2) + z(-2)*f + z(-1)*g - h;
indexrange 1,2;
factor z;

```

%The integrability conditions:

```

integ(k) := d w(k) ^ w(1) ^ w(2);

```

```

clear a,b,c,f,g,h,w(k),integ(k);

```

%Problem:

%-----

```

%Calculate the PDE's for the generators of the d-theta symmetries of
%the Lagrangian system of the planar Kepler problem.
%(c.f. W.Sarlet, F.Cantrijn, Siam Review 23, 467, 1981;
%Verify that time translation is a d-theta symmetry and calculate the
%corresponding integral.

```

```

pform t=0,q(k)=0,v(k)=0,lam(k)=0,tau=0,xi(k)=0,et(k)=0,theta=1,f=0,
    l=0,glq(k)=0,glv(k)=0,glt=0;

```

```

tvector gam,y;

```

```

indexrange 1,2;

```

```

fdomain tau=tau(t,q(k),v(k)),xi=xi(t,q(k),v(k)),f=f(t,q(k),v(k));

```

```

l:=1/2*(v(1)**2+v(2)**2)+m/r$      %The Lagrangian

```

```

pform r=0;

```

```

fdomain r=r(q(k));

```

```

let @ (r,q 1)=q(1)/r,@ (r,q 2)=q(2)/r,q(1)**2+q(2)**2=r**2;

```

```

lam(k):=-m*q(k)/r;

```

%The force

```

gam:=@ t + v(k)*@ (q(k)) + lam(k)*@ (v(k))$

```

```

et(k) := gam _| d xi(k) - v(k)*gam _| d tau$

```

```

y :=tau*@ t + xi(k)*@ (q(k)) + et(k)*@ (v(k))$      %Symmetry generator

```

```

theta := l*d t + @ (l,v(k))*(d q(k) - v(k)*d t)$

```

```

factor @;

```

```
s := y |_ theta - d f$

glq(k):=0(q k)_|s;
glv(k):=0(v k)_|s;
glt:=0(t)_|s;

%Translation in time must generate a symmetry.
xi(k) := 0;
tau := 1;

glq k;
glv k;
glt;

%The corresponding integral is of course the energy.
integ := - y |_ theta;

clear l,lam k,gam,et k,y,theta,s,glq k,glv k,glt;
remfdomain r,f;

%Problem:
%-----
%Calculate the "gradient" and "Laplacian" of a function and the "curl"
%and "divergence" of a one-form in elliptic coordinates.

coframe e u=sqrt(cosh(v)**2-sin(u)**2)*d u,
      e v=sqrt(cosh(v)**2-sin(u)**2)*d v,
      e ph=cos u*sinh v*d ph;

pform f=0;

fdomain f=f(u,v,ph);

factor e,^;
on rat,gcd;
order cosh v, sin u;
%The gradient:
d f;

factor 0;
%The Laplacian:
# d # d f;

%Another way of calculating the Laplacian:
-#vardf(1/2*d f^#d f,f);

remfac 0;

%Now calculate the "curl" and the "divergence" of a one-form:
```

```
pform w=1,a(k)=0;

fdomain a=a(u,v,ph);

w:=a(-k)*e k;
%The curl:
x := # d w;

factor 0;
%The divergence:
y := # d # w;

remfac 0;
clear x,y,w,a;
remfdomain a,f;

%Problem:
%-----
%Calculate in a spherical coordinate system the Navier Stokes equations.

coframe e r=d r,e th=r*d th,e ph=r*sin th*d ph;
frame x;

fdomain v=v(t,r,th,ph),p=p(r,th,ph);

pform v(k)=0,p=0;

%We first calculate the convective derivative.

v := v(-k)*e(k)$

factor e; on rat;

cdv := 0(v,t) + (v(k)*x(-k)) |_ v - 1/2*d(v(k)*v(-k));

%next we calculate the viscous terms.

visc := nu*(d#d# v - #d#d v) + nus*d#d# v;

%finally we add the pressure term and print the components of the
%whole equation.

pform nast=1,nast(k)=0;

nast := cdv - visc + 1/rho*d p$

factor 0;

nast(-k) := x(-k) |_ nast;
```

```
clear v k, x k, nact k, cdv, visc, p;
remfdomain p, v;
```

%Problem:

%-----

%Calculate from the Lagrangian of a vibrating rod the equation of  
% motion and show that the invariance under time translation leads  
% to a conserved current.

```
pform y=0, x=0, t=0, q=0, j=0, lagr=2;
```

```
fdomain y=y(x,t), q=q(x), j=j(x);
```

```
factor ^;
```

```
lagr:=1/2*(rho*q*(y,t)**2-e*j*(y,x,x)**2)*d x^d t;
```

```
vardf(lagr,y);
```

%The Lagrangian does not explicitly depend on time; therefore the  
%vector field  $\partial_t$  generates a symmetry. The conserved current is

```
pform c=1;
```

```
factor d;
```

```
c := noether(lagr,y,0 t);
```

%The exterior derivative of this must be zero or a multiple of the  
%equation of motion (weak conservation law) to be a conserved current.

```
remfac d;
```

```
d c;
```

%i.e. it is a multiple of the equation of motion.

```
clear lagr,c;
```

%Problem:

%-----

%Show that the metric structure given by Eguchi and Hanson induces a  
%self-dual curvature.

%c.f. T. Eguchi, P.B. Gilkey, A.J. Hanson, "Gravitation, Gauge Theories  
% and Differential Geometry", Physics Reports 66, 213, 1980.

```
for all x let cos(x)**2=1-sin(x)**2;
```

```
pform f=0,g=0;
```

```
fdomain f=f(x), g=g(x);
```

```
coframe o(r)=f*d r,
o(theta)=(r/2)*(sin(psi)*d theta-sin(theta)*cos(psi)*d phi),
o(phi)=(r/2)*(-cos(psi)*d theta-sin(theta)*sin(psi)*d phi),
o(psi)=(r/2)*g*(d psi+cos(theta)*d phi);
```

```
frame e;
```

```
pform gammal(a,b)=1, curv2(a,b)=2;
antisymmetric gammal, curv2;
```

```
factor o;
```

```
gammal(-a,-b):=-(1/2)*( e(-a)_l(e(-c)_l(d o(-b)))
-e(-b)_l(e(-a)_l(d o(-c)))
+e(-c)_l(e(-b)_l(d o(-a))) )*o(c)$
```

```
curv2(-a,b):=d gammal(-a,b) + gammal(-c,b)^gammal(-a,c)$
```

```
factor ^;
```

```
curv2(a,b):= curv2(a,b)$
```

```
let f=1/g;
```

```
let g=sqrt(1-(a/r)**4);
```

```
pform chck(k,l)=2;
```

```
antisymmetric chck;
```

%The following has to be zero for a self-dual curvature.

```
chck(k,l):=1/2*eps(k,l,m,n)*curv2(-m,-n)+curv2(k,l);
```

```
clear gammal(a,b), curv2(a,b), f, g, chck(a,b), o(k), e(k);
remfdomain f, g;
```

%Problem:

%-----

%Calculate for a given coframe and given torsion the Riemannian part and  
%the torsion induced part of the connection. Calculate the curvature.

%For a more elaborate example see E.Schruefer, F.W. Hehl, J.D. McCrea,  
%"Exterior calculus on the computer: The REDUCE package EXCALC applied to  
%general relativity and to the Poincare gauge theory", to appear in  
%GRG Journal.

```
pform ff=0, gg=0;
```

```
fdomain ff=ff(r), gg=gg(r);
```

```
coframe o(4)=d u+2*b0*cos(theta)*d phi,
o(1)=ff*(d u+2*b0*cos(theta)*d phi)+d r,
o(2)=gg*d theta,
```

```
o(3)=gg*sin(theta)*d phi
with metric g=-o(4)*o(1)-o(4)*o(1)+o(2)*o(2)+o(3)*o(3);

frame e;

pform tor(a)=2, gwt(a)=2, gam(a,b)=1,
u1=0, u3=0, u5=0;

antisymmetric gam;

fdomain u1=u1(r), u3=u3(r), u5=u5(r);

tor(4):=0$

tor(1):=-u5*o(4)^o(1)-2*u3*o(2)^o(3)$
tor(2):=u1*o(4)^o(2)+u3*o(4)^o(3)$
tor(3):=u1*o(4)^o(3)-u3*o(4)^o(2)$

gwt(-a):=d o(-a)-tor(-a)$

%The following is the combined connection.
%The Riemannian part could have equally well been calculated by the
%RIEMANNCONX statement.

gam(-a,-b):=(1/2)*( e(-b)_|(e(-c)_|gwt(-a))
                    +e(-c)_|(e(-a)_|gwt(-b))
                    -e(-a)_|(e(-b)_|gwt(-c)) )*o(c);

pform curv(a,b)=2;
antisymmetric curv;
factor ^;

curv(-a,b):=d gam(-a,b) + gam(-c,b)^gam(-a,c);

showtime;
end;
```

