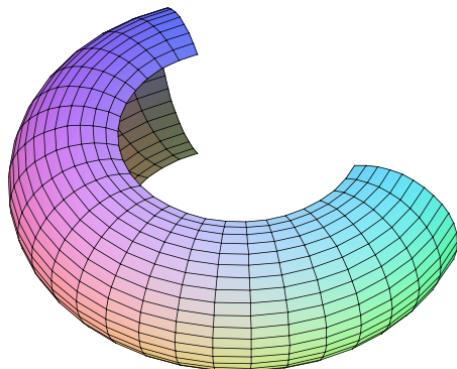


Exploring Math  with MAXIMA

# Classical Differential Geometry of Surfaces

with MAXIMA



Dr. Wolfgang Lindner  
[dr.w.g.Lindner@gmail.com](mailto:dr.w.g.Lindner@gmail.com)  
Leichlingen, Germany  
2023

CONTENTS	1
----------	---

## Contents

<b>1 Surfaces in Three Dimensions</b>	<b>5</b>
1.1 Definition - Surface . . . . .	5
1.2 Function Surfaces . . . . .	7
1.3 Level Surfaces . . . . .	11
1.4 Parametric Surfaces . . . . .	13
<b>2 The GAUSS Map</b>	<b>17</b>
2.1 Definition . . . . .	17
2.2 Examples of surface normals . . . . .	18
2.3 Exercises . . . . .	26
<b>3 The First Fundamental Form</b>	<b>29</b>
3.1 Definition . . . . .	30
3.2 Implementation and Examples . . . . .	31
3.3 Exercises . . . . .	33
<b>4 The Second Fundamental Form</b>	<b>36</b>
4.1 Definition . . . . .	37
4.2 Implementation and Examples . . . . .	38
4.3 Exercises . . . . .	42
<b>5 Shape Operator and WEINGARTEN Map</b>	<b>45</b>
5.1 Definition . . . . .	45
5.2 Examples . . . . .	46
5.3 Exercises . . . . .	48
<b>6 Scalar Curvature Measures</b>	<b>49</b>
6.1 Definition . . . . .	49
6.2 Examples . . . . .	50
6.3 Exercises . . . . .	59
<b>7 CHRISTOFFEL Symbols and RIEMANN Tensors</b>	<b>62</b>
7.1 Definition . . . . .	62
7.2 Example - $\Gamma$ and $R^i_{jkl}$ for the saddle surface $z = xy$ . . . . .	64
7.3 Example - $\Gamma$ and $R^i_{jkl}$ for the Helicoid . . . . .	71
7.4 Exercises . . . . .	77
<b>8 Intrinsic Curvature</b>	<b>80</b>
8.1 Definition . . . . .	80
8.2 Implementation and Examples . . . . .	80
8.3 Exercises . . . . .	84

<i>CONTENTS</i>	2
<b>9 RICCCI Tensor and GAUSS Theorema egregium</b>	<b>87</b>
9.1 Definition - RICCI curvature . . . . .	87
9.2 GAUSS's Theorema Egregium . . . . .	88
9.3 Implementation and Examples . . . . .	89
9.4 The sine-Gordon Equation . . . . .	92
9.5 The RICCI scalar of the SCHWARZSCHILD Spacetime Metric . . . . .	94
9.6 The RICCI scalar of the GÖDEL Spacetime Metric . . . . .	97
9.7 Exercises . . . . .	98
<b>10 The Fundamental Theorem</b>	<b>99</b>
10.1 The WEINGARTEN Equations . . . . .	99
10.2 The GAUSS Equations . . . . .	101
10.3 The CODAZZI-MAINARDI Equations . . . . .	104
10.4 The Fundamental Theorem of Surfaces Theory . . . . .	107
10.5 Exercises . . . . .	107
<b>11 Covariant Derivative</b>	<b>109</b>
11.1 Definition . . . . .	109
11.2 Implementation and Examples . . . . .	110
11.3 Exercise . . . . .	113
<b>12 Appendix: source code of diffgeoBox</b>	<b>115</b>
<b>13 Bibliography</b>	<b>117</b>

## Preface

This booklet would like to whet your appetite to immerse yourself into the world of Classical Differential Geometry. We do not reproduce the mathematical theory here, but want instead focus on typical examples and their computations with the help of the free CAS MAXIMA. The mathematical concepts and operations in Differential Geometry are often clumsy and uncomfortable and accompanied by heavy calculations, so MAXIMA as your computer algebraic companion should unburden you of calculations in this theme. We only cite some of the necessary underlying mathematical definitions and facts to be able to show the corresponding implementation of the concepts into the language of the computer algebra system MAXIMA.

For the mathematical description of Differential Geometry parallel with this script, I recommend the treatments by GALLOWAY, HECKMANN, HERTRICH-JEROMIN, HITCHIN, SHIFRIN and WHEELER to be found on the internet. These free online texts presents proofs of proposition and facts, that are only cited here. Recommended books are e.g. BANCHOFF/LEVITT, PRESSLEY or THORPE, mentioned in the bibliography.

The collection of the MAXIMA examples and exercise scripts in this booklet not only want to help the reader to dive into the classical Differential Geometry of surfaces, but also to become comfortable with the use of the CAS MAXIMA in this field. Therefore, to program the necessary concepts as simply and directly as possible, I limit myself to the use of build-in functions without the use of additional packages.

To run a MAXIMA script of this booklet no installation is necessary, *everything runs directly online*: a click on a link like • ▷ Click here in this text is enough to invoke MAXIMA<sup>on line</sup>. Of course, you can also copy the presented code to wxMAXIMAs input line and then press Shift-Enter to execute the code.

This script is based on a booklet, where I used the small free compact CAS EigenMath to do the computations directly online. This text is to be found at LINDNER [17].

There is a solution booklet with cross-checked calculations using Maple Vr4 and the packet `diffgeo.m` by RECKZIEGEL [25] to be found at LINDNER [18].

The author studied Differential Geometry from his dear academic teacher Prof. Dr. Peter DOMBROWSKI at the university of Cologne, Germany, in the 70ies of the last century in a suite of wonderful precise, crystal clear and inspiring lectures. At that time, illustrations in the lectures were hardly possible, mostly given with chalk sketches on the blackboard or hand sketches in the script. Calculations in the lectures exercises were tedious. This has now all changed for the better, this booklet try to demonstrate this with MAXIMA for the calculations and - occasionally - with CALC PLOT3D for some plots.

I dedicate this little book as a friendly reference to P. DOMBROWSKI. I also took some nice lectures by Prof. Dr. Helmut RECKZIEGEL, who later wrote a fine book on Differential Geometry with Maple.



Peter DOMBROWSKI

I would like to thank Viktor T. TOTH for his friendly help with the `ctensor` package.  
User comments are very welcome.

Wolfgang Lindner  
Leichlingen, Germany  
April 2023  
[Dr.W.G.Lindner@gmail.com](mailto:Dr.W.G.Lindner@gmail.com)

# 1 Surfaces in Three Dimensions

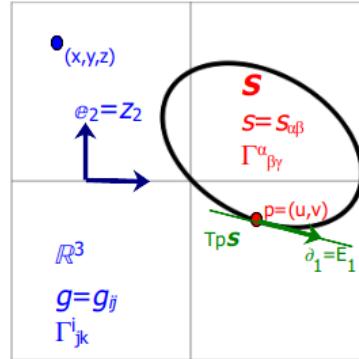
Differential Geometry make heavy use of (*partial*) differentiating functions. We start by quoting a definition what a *surface in space* is, cf. e.g. SHIFRIN [33, p.35]. Then we visit and explore some specimens and individuals in the zoo of surfaces and show how to express and inspect surfaces and their characteristics in MAXIMA.

## 1.1 Definition - Surface

Let  $U$  be an open<sup>1</sup> set in  $\mathbb{R}^2$ . Let  $X: U \rightarrow \mathbb{R}^3$  be a function on  $U$ .

We will use  $u, v$  as coordinates in our parameter region  $U$ , and  $x, y, z$  as coordinates in  $\mathbb{R}^3$ .

- a.  $X$  is called  $C^1$  iff  $X$  and its partial derivatives  $\frac{\partial X}{\partial u}$  and  $\frac{\partial X}{\partial v}$  exist and are continuous.
  - b.  $X$  is called *smooth* iff  $X$  is  $C^3$ . (We will assume all our functions  $X$  are  $C^3$ .)
  - c.  $X: U \rightarrow S \subset \mathbb{R}^3$  is called *parametrization of  $S$*  iff  $X$  is  $C^3$  and 1-to-1 and  $X_u \times X_v \neq 0$ , i.e.  $(X_u, X_v)$  are linear independent.
  - d.  $S \subset \mathbb{R}^3$  is called a *surface* iff  $S$  is connected and each point has a neighborhood that is (regularly) parametrized.
  - We will use subscripts a la ....<sub>u</sub> to indicate partial derivatives, as follows:
- $$f_u \equiv \frac{\partial f}{\partial u} \quad f_v \equiv \frac{\partial f}{\partial v} \quad f_{uu} \equiv \frac{\partial^2 f}{\partial u^2} \quad f_{uv} \equiv \frac{\partial^2 f}{\partial u \partial v}$$
- As a first preview on what follows in the forthcoming chapters and to give an 'advanced organizer' we present in one scene most of the objects and concepts we will discuss later in detail:



*Basic perception/mental image of space vs surface:*

$\mathbb{R}^3$ : the ambient **space** □.

$S$ : a **surface** ○ inside the ambient space.

$T_p S$ : tangential space at  $p$  on surface the  $S$ .

$e_2 = Z_2$  ..: basis vectors in ambient space, denoted  $e$  or  $Z$ .

$\partial_1 = E_1$  ..: basis in tangential space  $T_p S$ , denoted  $\partial$  or  $E$ .

---

<sup>1</sup>If in the following chapters  $U$  is not open, we could easily find an open set  $U'$  with  $U \subset U'$ .

- : a point with coordinates  $(x, y, z) \in \mathbb{R}^3$  in ambient space.
- : a point with coordinates  $(u, v) \in U$  on surface the  $S$ .

**Remark.** We agree on the following conventions and notations in order to distinguish between objects in space respectively on the surface if necessary. In doubt we use latin letters like  $i, j, k$  (running through 1,2,3) to index mathematical entities in ambient space  $V = \mathbb{R}^3$  and we use greek letters like  $\alpha, \beta$  (running through 1,2) to address mathematical objects in a surface  $S$ . Per example, the covariant metric tensor in ambient space  $g$  (in short: the space metric  $g$ ) is noted as  $g_{ij}$ , whereas the covariant metric tensor in surface (in short: the surface metric  $s$ ) may be noted as  $g_{\alpha\beta}$  or as  $s_{\alpha\beta}$ . If the context is clear and there is no doubt, we also use  $g_{ij}$  as notation for the surface metric - this is usual.

LEXICON	Math	MAXIMA
space coordinates $X = (x, y, z)$ or $x_i$	$x_i$	$X : [x,y,z]$
surface coordinates $U = (u, v)$ or $u_\alpha$	$u_\alpha$	$U : [u,v]$
space basis $e_1, e_2, e_3$	$e_i, i = 1, 2, 3$	$e1, e2, e3$
surface basis $\partial_1, \partial_2$ or $E_1, E_2$	$\partial_i$ or $E_i, i = 1, 2$	$E1, E2$
space metric $g = g_{ij}$	$g_{ij}$	$g; gij$

We start with some memories about surfaces of multivariate functions. We remember of graphs alias function surfaces, level sets and parameterizations. Function surfaces ('graphs') are preferred in advanced calculus, when it comes to partial derivatives and surface integrals, cf. [22] or [34, p.111, p.207 ff]. Parameterizations of surfaces are the main tool in Differential Geometry and we will focus later on that.

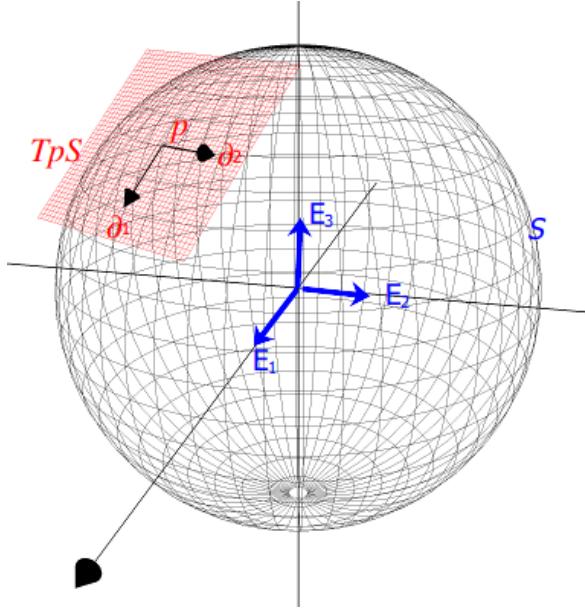
**Remark.** (A hint for the use of MAXIMA)

If you see a link like

- **Mark & Copy** the blue code lines. ▷ Then Click here and **Paste** it into the input box . Press .

you may use MAXIMA or e.g. MAXIMA<sup>on line</sup> or SAGEMATHCELL. If you use MAXIMA<sup>on line</sup> do not forget to *click into the Online form* to give it the focus. You have the focus, if the MAXIMA<sup>on line</sup> *Online frame*  change to **blue**. Please check, whether all input lines are pasted with the right **NEW LINE** ending! Otherwise correct the pasting inside the online frame so that the code is formatted exactly as in the blue template here in this booklet.

If you use wxMAXIMA you paste the code lines into the input region . The MAXIMA outputs in this text are the results of running wxMAXIMA on the iMac.



Visualization of the tangential plane  $T_pS$  at  $p$  on surface  $S$

Figure 2: with tangential basis vectors  $\partial_1$  and  $\partial_2$  and ambient space basis  $E_1, E_2, E_3$  in  $\mathbb{R}^3$ , cf. REJBRAND [26].

## 1.2 Function Surfaces

The graph of a function of two variables is a surface in space. The graph of function  $f(x, y)$  consists of all points  $(x, y, z)$  in space such that  $(x, y)$  is in the definition domain of the function and  $f(x, y) = z$ , i.e.

$$\text{graph}(f) := \{(x, y, z) \in \mathbb{R}^3 \mid z = f(x, y)\}$$

Function surfaces alias function *graphs* are used primarily in multivariate calculus books, cf. DINEEN [7] or MARDSEN [22].

**Example 1.** We plot  $\text{graph}(f)$  of the function  $f: \mathbb{R}^2 \rightarrow \mathbb{R}$  defined by  $f(x, y) = xy$  with CALC PLOT3D for an easy interactive plotting. Start CALC PLOT3D by clicking this link:  
▷ Open CALC PLOT3D .

Then we only have to override the given function term  $z = 7xy/\exp(x^2 + y^2)$  with  $xy$  in the Graph menu:

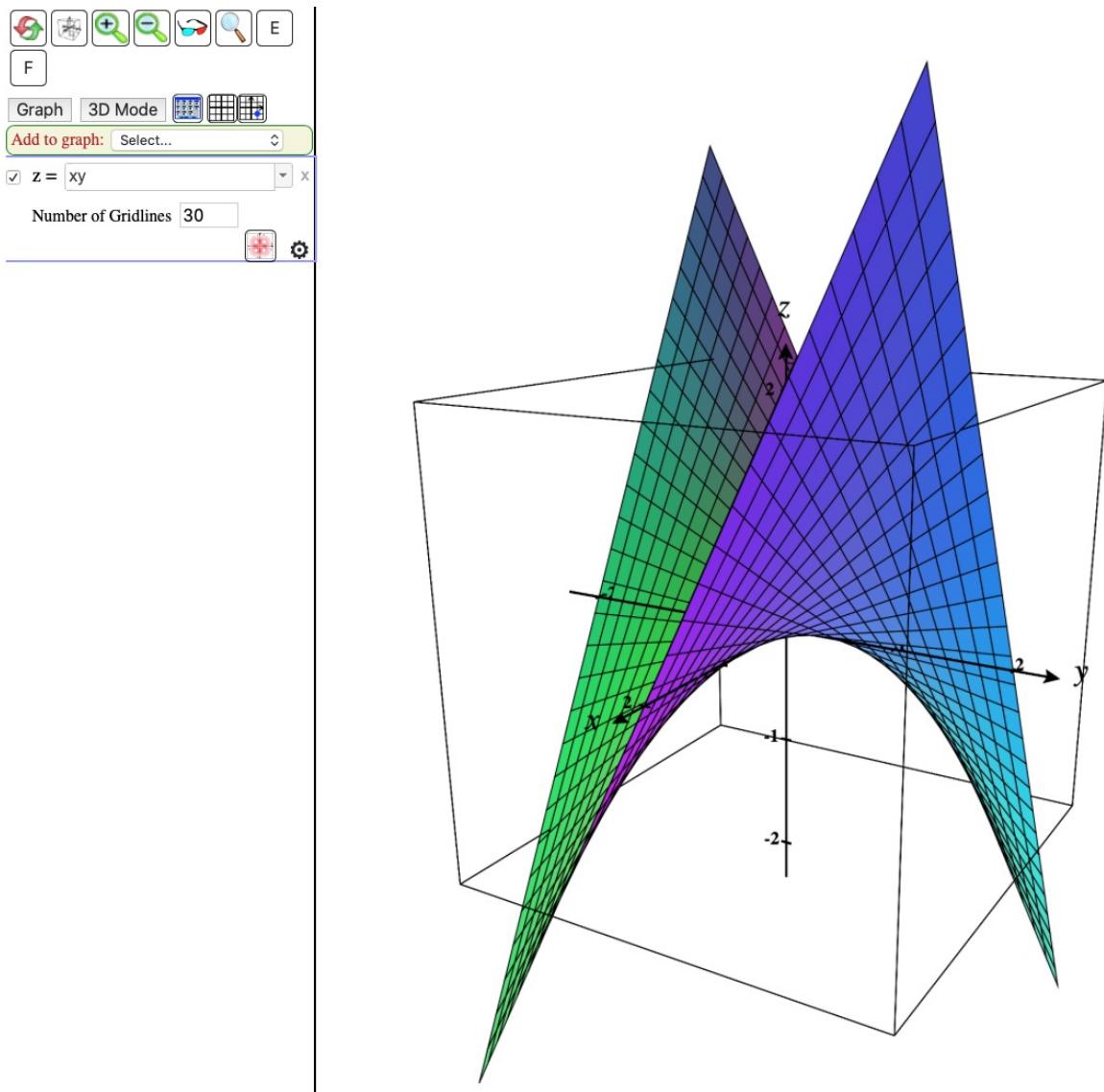
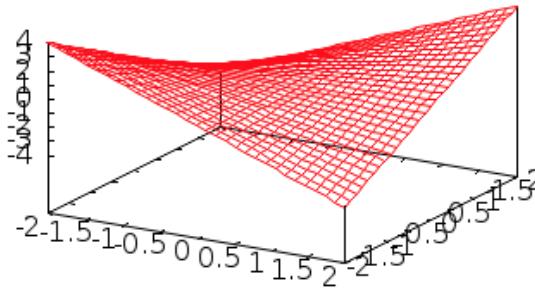


Figure 3: surface plot of  $\text{graph}(f)$  in  $\mathbb{R}^3$  of function  $f(x, y) = xy$  over rectangle  $U = [-2, +2] \times [-2, +2]$  in  $\mathbb{R}^2$  with CALC PLOT3D.

We plot the same function surface using Maxima's `draw3d` explicit with MAXIMA<sup>on line</sup>:

```
draw3d( color = red,
    explicit(x*y, x,-2,2, y,-2,2)) $
```



To use function  $f$  e.g. to calculate some points on the  $f$ -surface we invoke MAXIMA<sup>on line</sup>.

```
/* Maxima-on-line */
f(x,y):= x^2 -x + 1;
f(3,2);
f(a,b);
"the point at the right peek:";
f(2,2);
```

- ▷ click here , copy & paste the blue code lines into the input box . Press Click.
- Here is the result of a run of this little script in wxMAXIMA and MAXIMA<sup>on line</sup>:

```
(%i5) f(x,y):= x^2 -x + 1;
f(3,2);
f(a,b);
"the point at the right peek";
f(2,2);

(%o1) f(x,y):=x^2-x+1
(%o2) 7
(%o3) a^2-a+1
(%o4) the point at the right peek
(%o5) 3
```

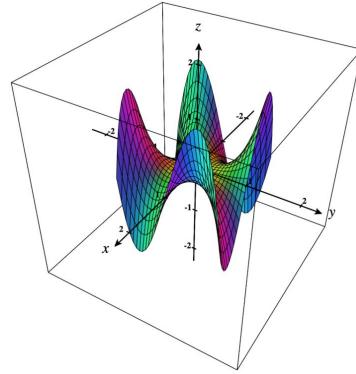
Maxima on line

```
(t11) f(x,y):= x^2 -x + 1;
(%o1) f(x,y) := x^2 - x + 1;
(t12) f(3,2);
(%o2) 7
(t13) f(a,b);
(%o3) a^2 - a + 1
(t14) "the point at the right peek";
(%o4) {} the point at the right peek()
(t15) f(2,2);
(%o5) 3
```

Left: wxMaxima – the input lines (%i5) and the collected corresponding output lines (%o1)....  
 Right: MAXIMA<sup>on line</sup> – the output line (%o..) follows immediately after the associated input line (%i..).

**Exercise 1.** Sketch the graph of  $f(x, y) = x - y + 2$ . For which coordinates  $x, y, z$  do you reach point  $p = (-2, 0, 0)$ ? C.f. [22, p.711]

**Exercise 2.** Reproduce the plot of the so-called *dog saddle* i.e. the function surface of  $f(x, y) = 4x^3y - 4xy^3$ , c.f. [22, p.722].



**Exercise 3.** Plot the function surface of  $f(x, y) = \frac{x^3 - 3x}{1+y^2}$ , c.f. [22, p.821].

**Exercise 4.** Plot the function surface of  $f(x, y) = \frac{\sin(\pi x)}{1+y^2}$ , c.f. [22, p.822].

**Exercise 5.** Plot the function surface of  $f(x, y) = \frac{x^2 - y^2}{x^2 + y^2}$ , c.f. [22, p.837].

**Exercise 6.** Plot the function surface of  $f(x, y) = \frac{y^3}{x^2 + y^2}$ , c.f. [22, p.837].

**Exercise 7.** Plot the function surface of  $f(x, y) = \frac{x^3 - 3xy^2}{x^2 + y^2}$ , c.f. [22, p.837].

### 1.3 Level Surfaces

Every graph is a level surface, so level sets generalize the concept of a graph.

Let  $f$  be a function of three variables and let  $c$  be a constant real number.

A *c-level surface* of  $f$ , denoted  $f^{-1}(c)$ , is the set of all allowable inputs pairs  $(x, y)$  with  $f(x, y) = c$ :

$$f^{-1}(c) := \{(x, y, z) \in \mathbb{R}^3 : f(x, y, z) = c\}$$

Level surfaces alias level sets of functions are primarily used by THORPE [37] and are also addressed in some detail in DINEEN[7].

**Example 2.** We plot the 1-level surface  $f^{-1}(1)$  of the function  $f: \mathbb{R}^3 \rightarrow \mathbb{R}$  defined by  $f(x, y, z) = x^2 + y^2 - z^2$ , i.e. the set  $\{(x, y, z) \in \mathbb{R}^3 : x^2 + y^2 - z^2 = 1\}$  with CALC PLOT3D.

In the **Add to graph:** menu we select **Implicit surface** and write in the input box

[Eq:  $x^2 + y^2 - z^2 = 1$ ].

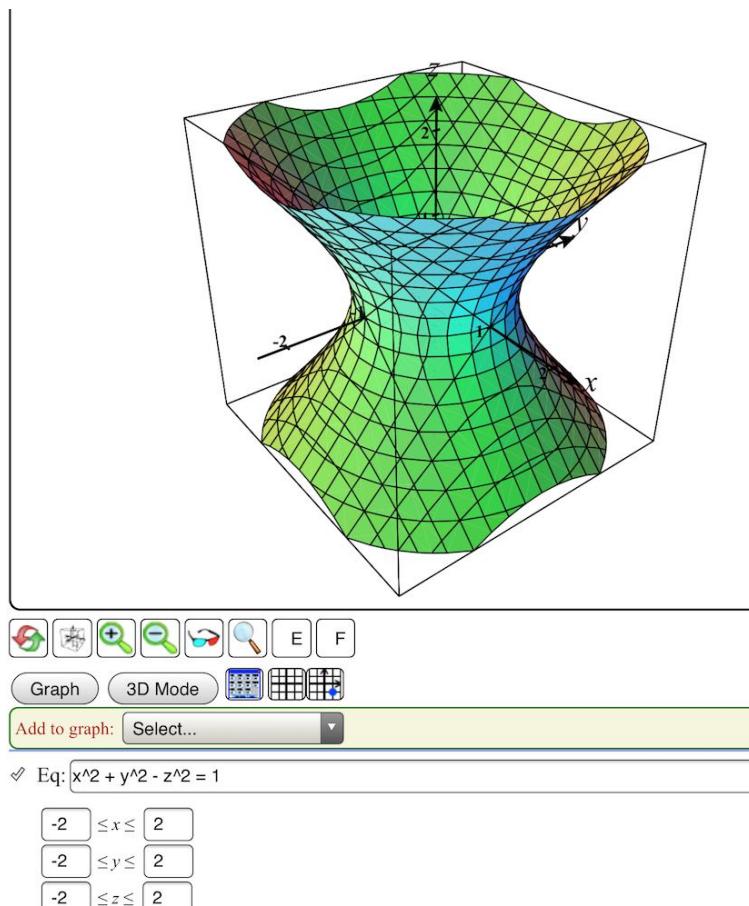


Figure 5: Level surface of  $x^2 + y^2 - z^2 = 1$  over the cube  $(x, y, z) \in [-2, +2] \times [-2, +2] \times [-2, +2]$ .

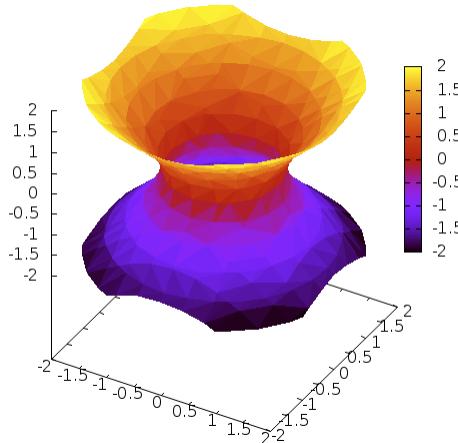
We check some points in  $\mathbb{R}^3$  for being members of the 1-level set of  $f$  with MAXIMA:

```
f(x,y,z) := x^2+y^2-z^2;
f(1,1,1);          /* (1,1,1) is in level surface */
f(sqrt(2),0,1);    /* (sqrt(2),0,1) is in level surface */
f(2,1,2);          /* (2,1,2) is in level surface */
```

▷ To open MAXIMA<sup>on line</sup> click here.

We plot the same level surface using Maxima's `draw3d implicit` with MAXIMA<sup>on line</sup>:

```
draw3d( color = red, enhanced3d = [z,x,y,z], dimensions = [500, 600],
implicit(x^2+y^2-z^2=1, x,-2,2, y,-2,2, z,-2,2)) $
```



**Exercise 8.** Let  $f(x, x, z) = x^2 + y^2 - z^2$ . Plot the level surface  $f(x, y, z) = -1$ . Cf. [22, p.722].

**Exercise 9.** Let  $f(x, y, z) = x^2 + 2y^2 + 3z^2 = 10$  be the ellipsoid realized as a level surface. Plot it.

**Exercise 10.** Let  $f(x, y, z) = xe^y - z$ . Plot the level surface  $f^{-1}(0)$ .

**Exercise 11.** Let  $f(x, y, z) = \sqrt{x} + \sqrt{y} + \sqrt{z}$ . Plot the level surface  $f^{-1}(4)$ .

## 1.4 Parametric Surfaces

Surfaces in 3-dimensional space are most described using a *parametric representation*. Each point  $p$  on a surface  $S$  is assigned a pair of parameter values  $u$  and  $v$  by a mapping  $X$ , i.e. a parametrization  $X$  maps a 2-dimensional region  $U$  of  $\mathbb{R}^2$  to part of a surface in  $\mathbb{R}^3$ .

- For example, the parametrization  $X: [0, 2\pi] \times [0, \pi] \rightarrow \mathbb{R}^3$  defined through

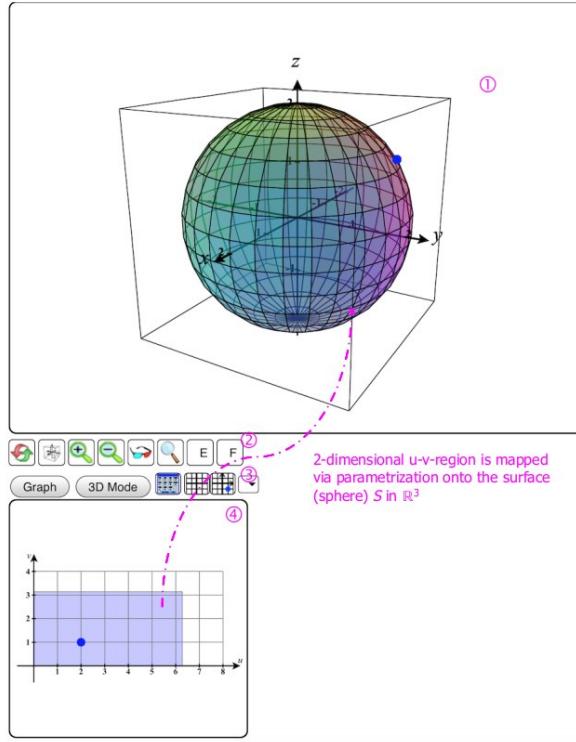
$$(u, v) \mapsto (2 \sin(v) \cos(u), 2 \sin(v) \sin(u), 2 \cos(v))$$

patches the 2-dimensional sphere  $S^2$ . One also denote this mapping by

$$x(u, v) = 2 \sin(v) \cos(u) \quad y(u, v) = 2 \sin(v) \sin(u) \quad z(u, v) = 2 \cos(v)$$

or shorter  $x = 2 \sin(v) \cos(u), y = 2 \sin(v) \sin(u), z = 2 \cos(v)$ .

This surface is visualized interactive by CALC PLOT3D by: invoke the menu via  
**≡ Examples > Parametric Surfaces > Sphere >** and click the trace pane t .



Parametric surface  $S$  by  $X$  with parameters  $u, v$ .

(1): the parametric surface  $X([0, 2\pi] \times [0, \pi])$ .

Figure 6: (2): some toggles for zoom, **E**dges, **F**aces etc.

(3): the trace pane to toggle  $(u, v)$  parameter region on/off.

(4): the parameter region  $[0, 2\pi] \times [0, \pi]$  is 'on'.

We implement this parametrization  $X$  using MAXIMA:

```
/* Maxima - DiffGeo01.txt */
X(u,v) := [2 * sin(v) *cos(u), 2* sin(v)* sin(u), 2* cos(v)];
X(u,v);
X(1,1);
float(X(1,1));
X(%pi/3,%pi/3);
```

▷ Open MAXIMA *on line* to RUN the code ... or ▷ SAGEMATHCELL, , if you do not use wxMAXIMA. wxMAXIMA output:

(%o1)	X(u,v):=[2 sin(v) cos(u),2 sin(v) sin(u),2 cos(v)]
(%o2)	[2 cos(u) sin(v),2 sin(u) sin(v),2 cos(v)]
(%o3)	[2 cos(1) sin(1),2 sin(1) <sup>2</sup> ,2 cos(1)]
(%o4)	[0.9092974268256818,1.416146836547142,1.08060461173628]
(%o5)	[\frac{\sqrt{3}}{2},\frac{3}{2},1]

In case you want to check, if  $u$  and  $v$  are admissible for input in the parametrization formula, you may test it in (1) before the calculation is done in (2) and otherwise through out an error message in (3):

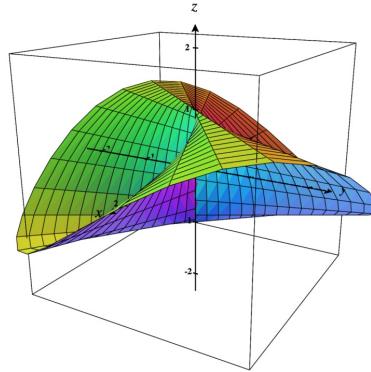
```
X(u,v) := if      ((0 <= u) and (u <= 2*%pi))           /*(1)*/
            or ((0 <= v) and (v <= %pi))
        then ([2 *sin(v)*cos(u), 2*sin(v)*sin(u), 2*cos(v)]) /*(2)*/
        else print("u or v not allowed." );                      /*(3)*/

float(X(1,1));
X(-%pi/3,%pi/3);
```

- Remark.**
1. Parametrization of surfaces is the most general concept to describe surfaces in space, cf. [7, p.100]. A parametrization is also called a surface *patch* or a *chart*. More information is found e.g. in [24, p.59 ff], [37, p.121 ff] or [23, p.1 ff] or [33, p.34].
  2. Because level surfaces and parametric surfaces are *locally* the same, cf. [37, p.121 ff], we use in the following mostly surface parametrizations.
  3. THORPE [37, p.110] gives the definition: A *parametrized 2-surface* in  $\mathbb{R}^3$  is a *smooth* map  $X: U \rightarrow \mathbb{R}^3$ , which is *regular*<sup>2</sup>. Herein  $U$  is a connected open set in  $\mathbb{R}^2$ .

---

<sup>2</sup>i.e. the JACOBI matrix of all partial derivatives is non-singular, which allows to have locally non-zero normal vector fields, see below and §1.1. a.–d.

**Exercise 12.** (PLÜCKER's Conoid)

- Identify PLÜCKER's Conoid in the [Parametric Surface](#) list of CALC PLOT3D.
- What is the parametric region for  $u, v$  ?
- Give the definition of the parametrization  $X$ .
- Write down the parametrization  $X$  in MAXIMA and do some tests.

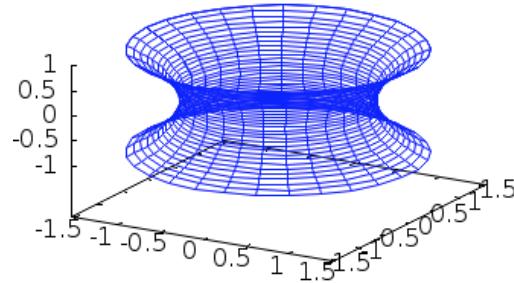
**Exercise 13.** (Catenoid) The *catenoid* is the parametric surface defined by the following parametric equations:

$$X(u, v) = \begin{bmatrix} c \cosh \frac{v}{c} \cos u \\ c \cosh \frac{v}{c} \sin u \\ v \end{bmatrix}$$

where  $u \in [-\pi, \pi]$  and  $v \in \mathbb{R}$  and  $c$  is a non-zero real constant.

- Plot the catenoid for  $c = 1$  using CALC PLOT3D.
- Here is how to easily plot the catenoid using MAXIMA's `draw` command in MAXIMA *on line*, try it:

```
draw3d( parametric_surface(
    cosh(v)*cos(u),cosh(v)*sin(u),v,
    u,-%pi,%pi, v,-1,1) )$
```



**Exercise 14.** (Helicoid) The *helicoid* is the parametric surface defined by

$$X(u, v) = (v \cos(\alpha u), v \sin(\alpha u), u)$$

with patch region  $u \in [-\pi, \pi]$ ,  $v \in \mathbb{R}$  and  $\alpha$  is a non-zero real constant. Plot the helicoid surface for  $\alpha = 1$  using CALC PLOT3D and Maxima.

**Exercise 15.** (SCHERK's surface) The SCHERK surface is the parametric surface defined by

$$X(u, v) = \log \left( \frac{\cos(u)}{\cos(v)} \right)$$

where  $(u, v) \in (-\frac{\pi}{2}, +\frac{\pi}{2}) \times (-\frac{\pi}{2}, +\frac{\pi}{2}) \subset \mathbb{R}^2$ .

Plot the SCHERK surface using CALC PLOT3D or Maxima.

**Exercise 16.** (ENNEPER's surface) The ENNEPER surface is the parametric surface defined by  $X(u, v) = (u - \frac{1}{3}u^3 + uv^2, v - \frac{1}{3}v^3 + vu^2, u^2 - v^2)$  with patch region  $u \in [-\pi, \pi], v \in \mathbb{R}$ . Plot the ENNEPER surface using CALC PLOT3D or Maxima.



Let's sum up:

LEXICON	<i>Math</i>	MAXIMA
<i>function</i> surface for $f: U \subset \mathbb{R}^2 \rightarrow \mathbb{R}$ :	$f(U) \subset \mathbb{R}^3$	$f(x,y)$
<i>level</i> surface for $f: U \subset \mathbb{R}^3 \rightarrow \mathbb{R}$ :	$f^{-1}(c) \subset \mathbb{R}^3$	$f(x,y,z)=c$
<i>parametric</i> surface for $X: U \subset \mathbb{R}^2 \rightarrow \mathbb{R}^3$ :	$X(U) \subset \mathbb{R}^3$	$X(u,v)$

There is no commonly accepted notation for a parametrization<sup>3</sup> and their partial derivatives, so we present this

LEXICON	<i>Math</i>	MAXIMA
Parametrization ; partial derivative w.r.t. $u$		
in BANCHOFF[2]:	$\vec{X}(u, v) ; \vec{X}_u$	$X(u,v) ; \text{diff}(X(u,v),u)$
in DINEEN[7]:	$\phi(u, v) ; \phi_u$	$\text{Phi}(u,v) ; \text{diff}(\text{Phi}(u,v),u)$
in GRINFELD[11]:	$S_\alpha ; \frac{\partial R}{\partial S_\alpha}$	$R(S1,S2) ; \text{diff}(R(S1,S2),S1)$
in HECKMANN [13], WHEELER [46]:	$\mathbf{r}(u, v) ; \mathbf{r}_u$	$\mathbf{r}(u,v) ; \text{diff}(\mathbf{r}(u,v),u)$
in PRESSLEY[24]:	$\sigma(u, v) ; \sigma_u$	$\text{sigma}(u,v) ; \text{diff}(\text{sigma}(u,v),u)$
in THORPE[37]:	$\varphi(u, v) ; \varphi_u$	$\text{phi}(u,v) ; \text{diff}(\text{phi}(u,v),u)$
in SHIFRIN[33]:	$x(u, v) ; x_u$	$\mathbf{x}(u,v) ; \text{diff}(\mathbf{x}(u,v),u)$

GRINFELD calls the parameters  $(u, v)$  the Surface coordinates, denoted  $(S1, S2)$ .

PRESSLEY denotes the parametrization  $\sigma$  after the greek lower case letter  $s$  for Surface.

HECKMANN and others denote the parametrization  $\mathbf{r}$  after the position (direction) vector pointing to a point  $R$  on the surface ... we follow the notation in BANCHOFF and use  $X$ .



We should now have a solid feeling for what a 2D surface is.

---

<sup>3</sup>Often  $\phi$  (=greek letter  $p$  for parameter) is used to denote a parameterization. We use  $X$  instead.

## 2 The GAUSS Map

We quote THORPE [37, p.31 ff], see also BANCHOFF [2, p.173] or SHIFRIN [33, p.45].

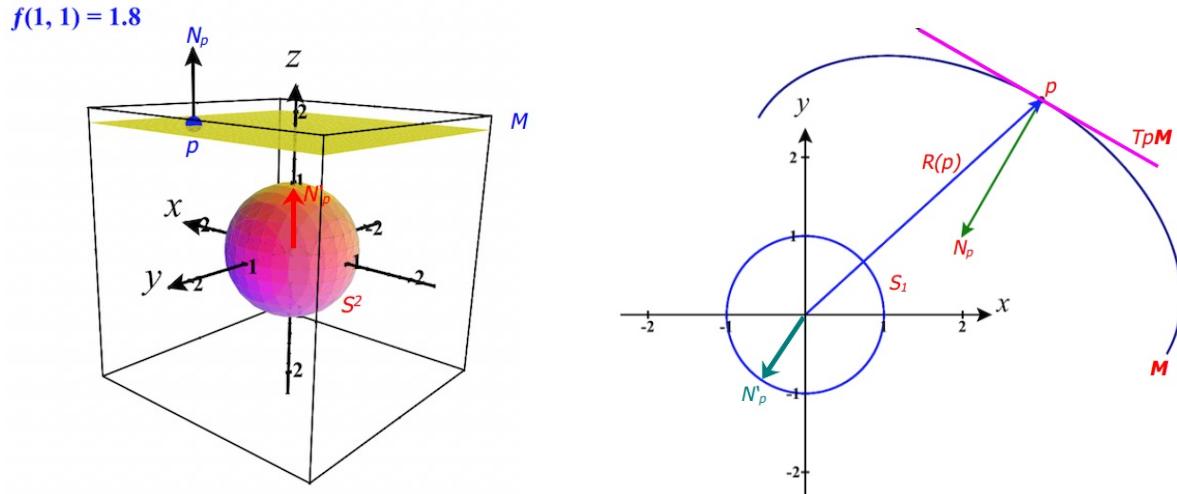
### 2.1 Definition

An *oriented* 2-surface  $S$  in  $\mathbb{R}^3$  is a 2-surface together with a smooth unit normal vector field  $N$  on  $S$ . The function  $N: S \rightarrow \mathbb{R}^3$  maps  $S$  into the unit sphere  $S^2$ . Think  $N: S \rightarrow S^2$  as map which assigns to each point  $p \in S$  the point in  $\mathbb{R}^3$  obtained by translating the unit normal vector  $N_p$  to the origin.

This map  $N$  is called the **GAUSS map**.

**Remark.** This concept has three important consequences:

1. The Gauss map  $N$  allows for an *intuitive visual feeling* how (much) the surface is '*curved*'. Look at Fig. 7! We will comment about it in a moment.
2. The *exterior* normal vector  $N$  allows for *explicit calculation formulas* of the *curvature* of a surface - with MAXIMA too. This will be a main theme of this booklet.
3. The normal  $N$  allows for the calculation and visualization of the tangent plane  $T_p S$  at a point  $p$  on the surface  $S$ . We will show how.



**Left:** Image of GAUSS map in  $S^2$  of a plane  $M$ . The blue normal vector  $N_p$  at  $p$  in plane  $M$  is mapped to the red vector  $N'_p$  in sphere  $S^2$ .

**Figure 7: Right:** Image of GAUSS map in  $S^1$  of a curve (part of ellipse, a 1D surface)  $M$ . The normal vector  $N_p$  at  $p$  at tangential 'plane'  $T_p M$  is mapped to the green vector  $N'_p$  in sphere  $S^1$ .

**Comment.** We interpret fig.7.

**Left:** The surface  $M$  is a plane in  $\mathbb{R}^3$ . At each point  $p$  of  $M$  the normal points in the same direction, i.e. the normal field is constant on the whole of  $M$ . Ergo the image of the GAUSS map remains the same constant normal  $N'_p$  on the sphere! This shows that the normal field does not change its value - the plane **is not 'curved'**. Using CALC PLOT3D we can move resp. trace the normal on the u-v-region and verify visually that  $N$  is constant and the coordinates of  $N$ , which are displayed at the green message pane above, remain the same.

**Right:** The 1D 'surface'  $M$  is part of an ellipse in  $\mathbb{R}^2$ . At the endpoint  $p$  of the position vector  $R(p)$  the green exterior normal  $N_p$  is perpendicular to the tangent plane  $T_p M$  of the surface  $M$ . If you hike along  $M$  in direction of the letter ' $y$ ' the normal rests perpendicular to the changing  $T_p M$ 's, but change its direction (and magnitude) always faster! Ergo  $N$  is not constant on  $M$  and its images  $N'_p$  builds a part of the sphere  $S^1$ : therefore the surface  $M$  **is 'curved'**. Using CALC PLOT3D we can trace the normal  $N$  on the u-v-region and verify visually that  $N$  is permanently changing its direction and also its coordinates, which are displayed at the CALC PLOT3D's green message pane above. This explains en passent what the concept '*field*' means ...

You should now have a first intuitive feeling what the *curvature* of a surface is.

LEXICON	<i>Math</i>	MAXIMA
<i>exterior surface normal vector field</i> :	$N$	$N$
<i>exterior surface normal vector at p</i> :	$N_p$	$N(p)$

## 2.2 Examples of surface normals

In the following we give examples of normal vectors (fields) of surfaces  $S$ , using all three representations - normals on function surfaces, on level surfaces and on parametric surfaces. We also demonstrate how to calculate the corresponding tangential space  $T_p S$  using the normal vector at a specific point  $p$ .

### 2.2.1 Normal on a Function Surface

If we parameterize  $S = \text{graph}(f)$  using the parameterization  $X(u, v) = (u, v, f(u, v))$ , then we can always choose

$$\textcolor{violet}{N} := \frac{(-f_u, -f_v, 1)}{\sqrt{1 + f_u^2 + f_v^2}} \quad (2.1)$$

as unit normal (vector field)  $N$ . Remember:  $f_u \equiv \frac{\partial f}{\partial u}$ .

**Example 3.** Let  $f: \mathbb{R}^2 \rightarrow \mathbb{R}$  defined by  $f(x, y) = x^2 + y^2 - 4x - 6y + 13$ , cf. [22, p.714].

- a. Calculate the unit normal  $N$  at  $p = (4, 3)$ .
- b. Plot the function surface with unit normal  $N$ .

*Solution:* We use MAXIMA.  $N$  is calculated by means of the differential operator `diff(f,x)`, which returns the partial derivative of  $f$  with respect to  $x$ .

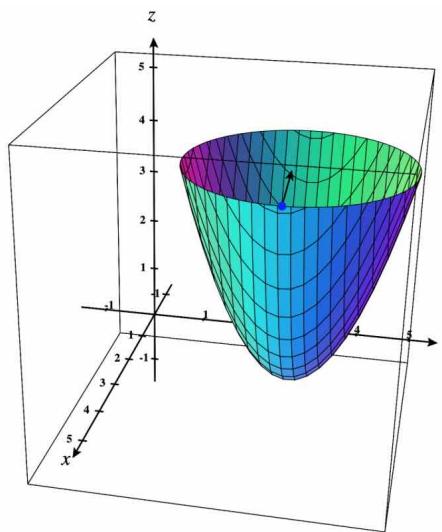
```
/* Maxima - DiffGeo02.txt */
f(u,v) := u^2+v^2-4*u-6*v+13;
f(u,v);
p : [4,3];

fu : diff(f(u,v),u);           /* partial derivative of f with respect to u */
fv : diff(f(u,v),v,1);

N : [-fu,-fv,1]/sqrt(1+fu^2+fv^2) ;
Np: subst([u=4,v=3],N);
```

▷ Open MAXIMA *on line* to RUN the code. or ▷ SAGEMATHCELL, if you do not use wxMAXIMA.  
wxMAXIMA output:

(%)1	$f(u, v) := u^2 + v^2 + (-4)u + (-6)v + 13$
(%)2	$v^2 - 6v + u^2 - 4u + 13$
(p)	[4, 3]
(fu)	$2u - 4$
(fv)	$2v - 6$
(N)	$\left[ \frac{4 - 2u}{\sqrt{(2v - 6)^2 + (2u - 4)^2 + 1}}, \frac{6 - 2v}{\sqrt{(2v - 6)^2 + (2u - 4)^2 + 1}}, \frac{1}{\sqrt{(2v - 6)^2 + (2u - 4)^2 + 1}} \right]$
(Np)	$\left[ -\frac{4}{\sqrt{17}}, 0, \frac{1}{\sqrt{17}} \right]$



**Comment.** In (N) we calculate the unit normal  $N$  using the given formula (2.1). With `subst(..)` we put  $u = 4$  and  $v = 3$  in the return value for  $N$ . We get

$$N_p = \left( \frac{-4}{\sqrt{17}}, 0, \frac{1}{\sqrt{17}} \right)$$

The plot of CALC PLOT3D shows a *paraboloid of revolution* with unit normal at  $p = (4, 3)$  pointing into the interior of the surface parallel to the  $u = x$  axis with a tiny elevation of  $-0.24$  in the direction of the  $z$ -axis. The approx. value of  $N_p$  is also displayed in the message pane of CALC PLOT3D as  $\vec{N}(4, 3) = -0.9701\hat{i} + 0.2425\hat{k}$ .

**Exercise 17.** Verify the formula for  $N$  using the given parameterization and the general formula for parametric surfaces in 2.2.3.

**Exercise 18.** (parabolic cylinder) Sketch the function surface given by  $z = -y^2$ , cf. [22, p.714]. Calculate and display the unit normal  $N$  at  $p = (1, 2)$ .

Trace  $N_p$  using a suited u-v-region and watch the changing  $N$ .

**Exercise 19.** (volcano) Sketch the function surface given by

$$z = (x^2 + 3y^2) \cdot e^{1-x^2-y^2}$$

over the u-v-domain  $[-2, +2] \times [-2, +2]$ , cf. [22, p.717].

Calculate and display the unit normal  $N$  at  $p = (1, 1)$ .

### 2.2.2 Normal on a Level Surface

It is well known from calculus, cf. e.g. [37, p.8], that the gradient of a function  $f$  of three variables is orthogonal to the tangential plane  $T_p S$  at that point, where the gradient is calculated. Let  $p$  lie on the level surface  $S$  defined by  $f(x, y, z) = c \in \mathbb{R}$ , then  $\text{grad}(f, p)$  is perpendicular ("normal") to  $T_p S$ , i.e.  $\text{grad}(f, p) \perp T_p S$ .

Therefore, if  $f$  is the defining function for a level surface  $S = f^{-1}(c)$  we always choose the unit normed gradient<sup>4</sup>  $\text{grad}(f)$  of  $f$  as the canonical unit normal vector field  $N$ :

$$\textcolor{violet}{N} := \frac{\text{grad}(f)}{\|\text{grad}(f)\|} \quad (2.2)$$

**Example 4.** Let  $f(x, y, z) = x^2 + y^2 - z^2$ . Calculate  $\text{grad}(f, p)$  for  $p = (0, 0, 1)$ .

Plot this gradient at the level surface  $f(x, y, z) = -1$ . c.f. [22, p.806].

*Solution:*  $\text{grad}(f)$  is first defined in (1) and then calculated in (2) using the differential operator `diff(f,x)`, which returns the partial derivative of  $f$  with respect to  $x$ .

```
/* MAXIMA - DiffGeo03.txt */
f(x,y,z) := x^2 + y^2 - z^2;
p : [0,0,1];
define( gradf(x,y,z),          /* (1) vector of all partial derivates */
        [diff(f(x,y,z),x), diff(f(x,y,z),y), diff(f(x,y,z),z)]);
gradf(x,y,z);                  /* (2) */
gradfp : subst( [x=0, y=0, z=1], gradf(x,y,z)); /* (3) */
```

---

<sup>4</sup>We denote the gradient of  $f$  by  $\text{grad}(f)$  and do not choose the common notation  $\nabla f$ , because MAXIMA does not have the nabla symbol  $\nabla$  available.

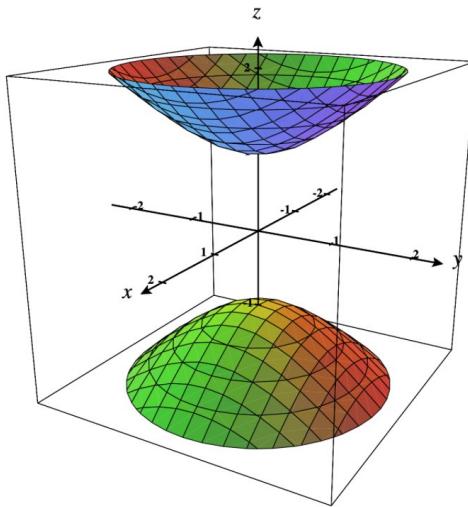
In (3) we calculate the value of  $\text{grad}f$  at the point  $p$ .

▷ Open MAXIMA<sup>on line</sup> or ▷ SAGEMATHCELL

MAXIMA output:

```
(%o1) f(x,y,z):=x^2+y^2-z^2
(%o2) [0,0,1]
(%o3) gradf(x,y,z):=[2 x, 2 y, -2 z]
(%o4) [2 x, 2 y, -2 z]
(%o5) [0,0,-2]
```

We plot the hyperboloid of two sheets using CALC PLOT3D. Plot it with MAXIMA.



**Exercise 20.** Calculate the unit normal  $N$  at  $p$  for the surface  $M$  in fig.7.left.

**Exercise 21.** Let  $f(x, y, z) = x^2 + 2y^2 + 3z^2 = 10$  be the ellipsoid realized as a level surface.

- a. Calculate a unit normal vector at each of the points  $p_1 = (\sqrt{10}, 0, 0)$ ,  $p_2 = (-\sqrt{10}, 0, 0)$  and  $q_1 = (1, 0, \sqrt{3})$ ,  $q_2 = (-1, 0, -\sqrt{3})$ .
- b. Do these normals point to the inside or outside of the ellipsoid?
- c. Plot the complete scene with CALC PLOT3D and verify your findings, cf. [22, p.807].

**Example 5.** (Example 2.3 of [7, p.18]) Let  $S$  denote the set of all points in  $\mathbb{R}^3$  which satisfy the equation  $x^2 + 2y^2 - 5z^2 = 1$ . Find the tangent space  $T_pS$  and the normal  $N_p$  at the point  $p = (2, -1, 1)$  on  $S$ .

*Solution:* Our ansatz is:

1. Calculate the normal vector field  $N$  of  $S$ . In this case  $N$  must not be of unit length.
2. Calculate  $T_pS$  using the fact that  $N \perp T_pS$ , i.e.  $T_pS: (X - p) \bullet N = 0$  for all  $X \in T_pM$ . We use MAXIMA.

```

/* Maxima - DiffGeo04.txt */
f(x,y,z) := x^2+2*y^2-5*z^2;
f(2,-1,1);           /* check if p in S, ok */
X : [x,y,z];         /* coordinates in ambient space R^3 */
p : [2,-1,1];
define(
    gradf(x,y,z),      /* -- grad is the normal vector field */
    [diff(f(x,y,z),x), diff(f(x,y,z),y), diff(f(x,y,z),z)]);
gradf(x,y,z);
gradfp : subst( [x=p[1], y=p[2], z=p[3]], gradf(x,y,z));
TpS : (X-p).gradfp; /* .=dot product; must be zero, because N_|_TpS */
expand(TpS);

```

- ▷ To open MAXIMA<sup>on line</sup> or ▷ SAGEMATHCELL

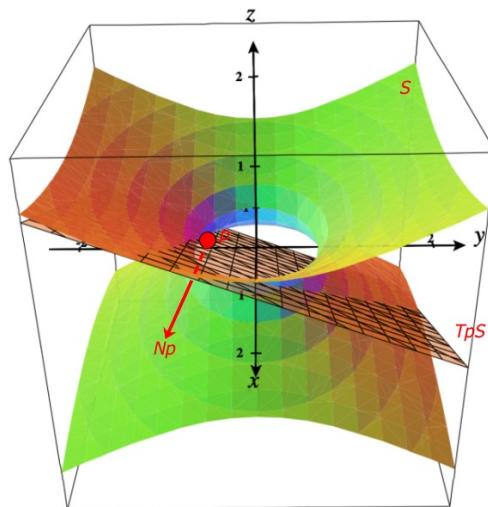
MAXIMA output:

```

(%o1) f(x,y,z):=x^2+2 y^2+(-5) z^2
(%o2) 1
(%o3) [x,y,z]
(%o4) [2,-1,1]
(%o5) gradf(x,y,z):=[2 x,4 y,-10 z]
(%o6) [2 x,4 y,-10 z]
(%o7) [4,-4,-10]
(%o8) -10 (z-1)-4 (y+1)+4 (x-2)
(%o9) -10 z-4 y+4 x-2

```

Because  $4x - 4y - 10z - 2 = 0$ , we resume that  $T_pS: 2x - 2y - 5z = 1$ . Let's plot the whole scene using CALC PLOT3D. Plot it with MAXIMA.



**Remark.** The normal  $N_p$  was added by hand ♡

### 2.2.3 Normal on a Parametric Surface

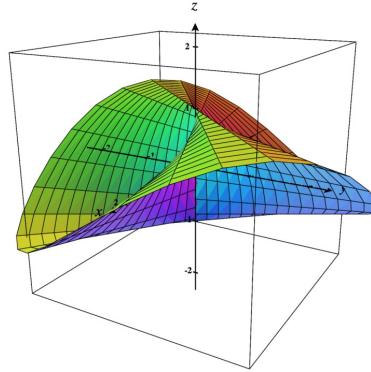
For a parameterized surface  $X: U \rightarrow \mathbb{R}^3$  with  $S = X(U)$  we always choose

$$N := \frac{X_u \times X_v}{\|X_u \times X_v\|} \quad (2.3)$$

as unit normal (vector field). Herein  $X_u$  is the partial derivative of  $X$  w.r.t. the first parameter  $u$ . It is well known that  $N$  is orthogonal to the tangential plane  $T_p S$  at the point, where  $N$  is calculated, i.e.

$$N_p \perp T_p S$$

**Example 6.** (PLÜCKER's Conoid)



PLÜCKER's Conoid is the parametrized surface defined by the parametric equations

$X(u, v) = (v \cdot \cos(u), v \cdot \sin(u), 2 \cos(u) \cdot \sin(u))$ , where  $u \in [0, 2\pi]$  and  $v \in [-3, 3]$ .

Calculate the normal vector field  $N_{pluecker}$ .

Calculate the exterior surface unit normal  $N_p$  at  $p = (\pi, 1)$ .

*Solution:* We use MAXIMA.  $N$  is calculated by means of the differential operator `diff(X(u,v),u)` which returns the partial derivative of the parameterization  $X$  with respect to  $u$ . We define the *cross product*  $u \times v$  of two vectors  $u$  and  $v$  and the *length* (alias the *magnitude*) of a vector  $v$  as user defined functions.

```
/* Maxima - DiffGeo05.txt */
define( X(u,v) , [v*cos(u), v*sin(u), 2*cos(u)*sin(u)]);
X(u,v);
X(%pi,1); /* the point p in S */

Xu : diff(X(u,v),u); /* partial derivative w.r.t. u */
Xv : diff(X(u,v),v);

define( cross(v,w), [v[2]*w[3] - w[2]*v[3],
                     w[1]*v[3] - v[1]*w[3],
                     v[1]*w[2] - w[1]*v[2]]);
```

```

mag(v) := sqrt(v.v);      /* magnitude of vector v */
N : cross(Xu,Xv);        /* exterior normal vector */
N : trigsimp(N);

magN : trigsimp(mag(N));
No   : N/magN;
Np   : subst([u=%pi, v=1], No);
fpprintprec : 5 $ 
float(Np);

```

MAXIMA output:

(N)	$[(2 - 4 \cos(u)^2) \sin(u), 4 \cos(u)^3 - 2 \cos(u), -v]$
(absN)	$\sqrt{v^2 + 16 \cos(u)^4 - 16 \cos(u)^2 + 4}$
(Np)	$[0, -\frac{\epsilon}{\sqrt{5}}, -\frac{1}{\sqrt{5}}]$
(%o428)	$[0.0, -0.89443, -0.44721]$

- ▷ To open MAXIMA *on line* or ▷ SAGEMATHCELL

**Comment.** In (Np) we have the normal vector field  $N(u, v)$ . In (%o428) we see the approximate value of the normal vector  $N_p = N(\pi, 1) \in \mathbb{R}^3$ .

**Example 7.** (Cylinder) The *cylinder* is the parametric surface defined by

$$X(u, v) = (a \cos(v), a \sin(v), u)$$

with patch region  $v \in [-\pi, \pi]$ ,  $u \in \mathbb{R}$  and  $a$  a non-zero real constant.

- Reproduce the plot of the cylinder in WOLFRAM [49] for  $a = 1$ .
- Calculate the normal vector field  $N$  for the cylinder.
- Calculate the exterior surface normal  $N_p$  at  $p = (1, \pi)$ .

*Solution to b. and c.* We use MAXIMA.

```

/* Maxima - DiffGeo06.txt */
"Cylinder";
define( X(u,v) , [1*cos(v), 1*sin(v), u]);
X(%pi,1);
Xu : diff(X(u,v),u);      /* partial derivative w.r.t. u */
Xv : diff(X(u,v),v);

define( cross(v,w), [v[2]*w[3] - w[2]*v[3],
                    w[1]*v[3] - v[1]*w[3],
                    v[1]*w[2] - w[1]*v[2]] );

```

```

mag(v) := sqrt(v.v);          /* magnitude of vector v */

N : cross(Xu,Xv);           /* exterior normal vector */
N : trigsimp(N);
magN : trigsimp(mag(N));    /* length of N */
No   : N/magN;              /* normed normal vector, i.e. |No|=1 */
Np   : subst([u=%pi, v=1], No);
fpprintprec : 5 $ 
float(Np);

```

MAXIMA output of  $N_o$  and  $N_p$ :

```

(magN) 1
(No)   [-cos(v), -sin(v), 0]
(Np)   [-cos(1), -sin(1), 0]
(%o467) [-0.5403, -0.84147, 0.0]

```

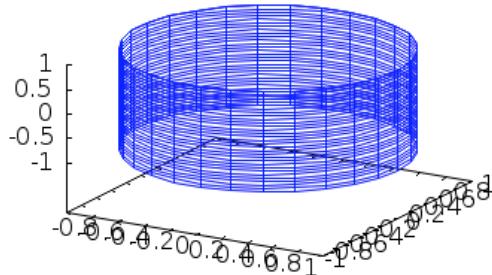
- ▷ Open MAXIMA<sup>on line</sup> or ▷ SAGEMATHCELL

Here is how to easy plot the cylinder using MAXIMA<sup>on line</sup>'s draw command:

```

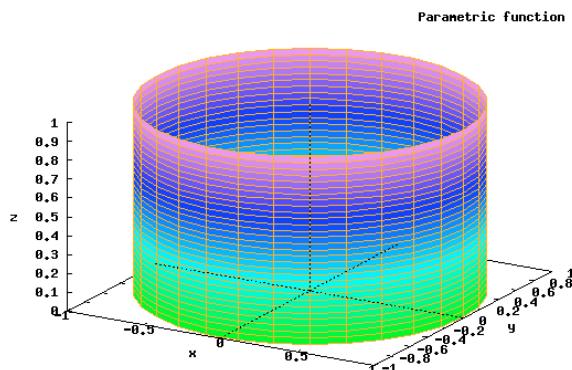
draw3d( parametric_surface(
      1*cos(v), 1*sin(v), u,
      v,-%pi,%pi, u,-1,1 )$)

```



or using wxMAXIMA

```
wxplot3d([1*cos(v), 1*sin(v), u],[v,0,2*pi],[u,0,1])$
```



### 2.3 Exercises

**Exercise 22.** (Cone) The *cone* is the parametric surface defined by

$$X(u, v) = (u \cos(v), u \sin(v), u)$$

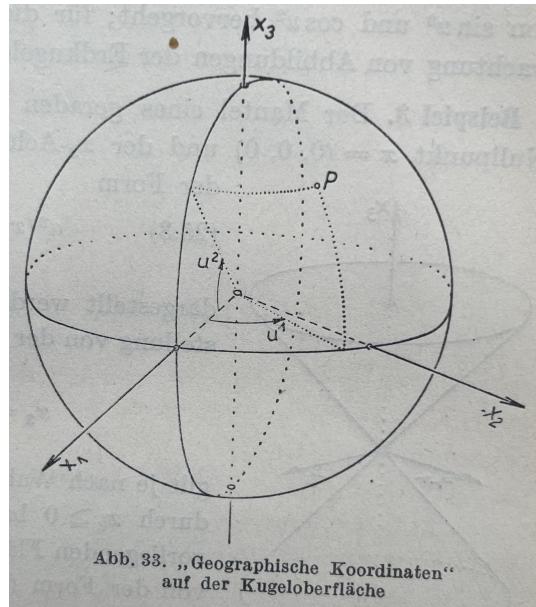
with patch region  $u \in [-\pi, \pi]$ ,  $v \in \mathbb{R}$ .

- a. Plot the cone using CALC PLOT3D or MAXIMA.
- b. Calculate the normal vector field  $N$  for the cone.
- c. Calculate the exterior surface normal  $N_p$  at  $p = (1, \pi)$ .

**Exercise 23.** (Sphere) The *r-sphere* is the parametric surface defined by

$$X(u, v) = (r \cos u \sin v, r \sin u \sin v, r \cos v)$$

with patch region  $u \in [0, 2\pi)$ ,  $v \in [0, \pi)$  and  $r$  a positive real constant.



Fine drawing of the sphere in an old book by KREYSZIG  
Figure 8: [15, p.93] with 'geographic coordinates' on the surface of  
the sphere. Lexicon:  $\begin{matrix} \text{Kreyszig: } u^1 u^2 \\ \text{here: } u v \end{matrix}$

- a. Plot the sphere using CALC PLOT3D or MAXIMA where  $r = 1$ .
- b. Calculate the normal vector field  $N$  for the unit sphere i.e.  $r = 1$ .
- c. Calculate the exterior surface normal  $N_p$  at  $p = (\pi, \pi)$ .

**Exercise 24.** (Helicoid) The *helicoid* is the parametric surface defined by

$$X(u, v) = (v \cos(\alpha u), v \sin(\alpha u), u)$$

with patch region  $u \in [-\pi, \pi]$ ,  $v \in \mathbb{R}$  and  $\alpha$  a non-zero real constant.

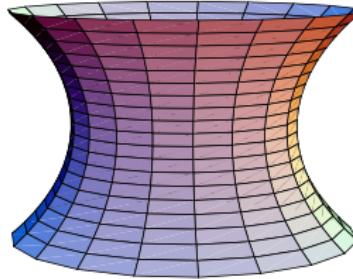
- a. Plot the helicoid for  $\alpha = 1$ .
- b. Calculate the normal vector field  $N$  for the helicoid.
- c. Calculate the exterior surface normal  $N_p$  at  $p = (\pi, 1)$ .

**Exercise 25.** (Catenoid) The *catenoid* is the parametric surface defined by the following parametric equations:

$$X(u, v) = \begin{bmatrix} c \cosh \frac{v}{c} \cos u \\ c \cosh \frac{v}{c} \sin u \\ v \end{bmatrix}$$

where  $u \in [0, 2\pi)$  and  $v \in \mathbb{R}$  and  $c$  is a non-zero real constant.

- a. Reproduce the plot of the *Catenoid* surface in WOLFRAM [47].



- b. Calculate the normal vector field  $N$  for the catenoid.<sup>5</sup>
- c. Calculate the exterior surface normal  $N_p$  at  $p = (\pi, 1)$ .

**Exercise 26.** (SCHERK's surface) The SCHERK surface is the 2D parametric surface defined by

$$X(u, v) = (u, v, \ln(\frac{\cos v}{\cos u}))$$

where  $u, v \in [0, 2\pi)$ .

- a. Plot the SCHERK surface.
- b. Calculate the normal vector field  $N$  for the SCHERK surface.
- c. Calculate the exterior surface normal  $N_p$  at  $p = (\pi, 1)$ .

---

<sup>5</sup>WOLFRAM gives  $N(u, v) = (-\cos(u)\operatorname{sech}(v/c), -\operatorname{sech}(v/c)\sin(u), \tanh(v/c))$ , cf. <https://mathworld.wolfram.com/HyperbolicSecant.html>.

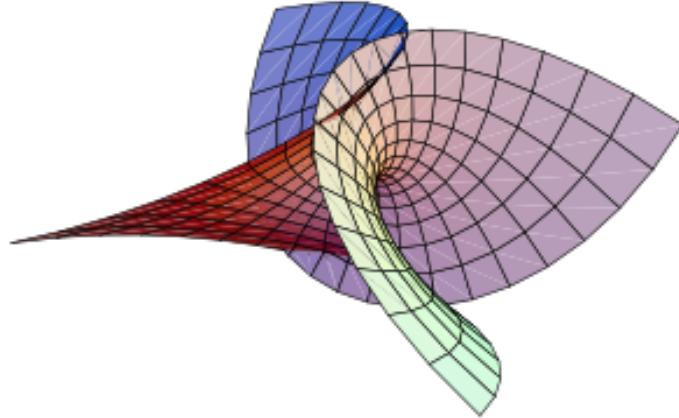
To get the concrete normal in c. you don't have to try to get exactly this result.

**Exercise 27.** (ENNEPER's surface) The ENNEPER surface is the 2D parametric surface defined by

$$X(u, v) = \left( u - \frac{1}{3}u^3 + uv^2, v - \frac{1}{3}v^3 + vu^2, u^2 - v^2 \right)$$

with patch region  $u \in [-\pi, \pi]$  and  $v \in \mathbb{R}$ .

- a. Reproduce the plot of ENNEPER surface in WOLFRAM [50].



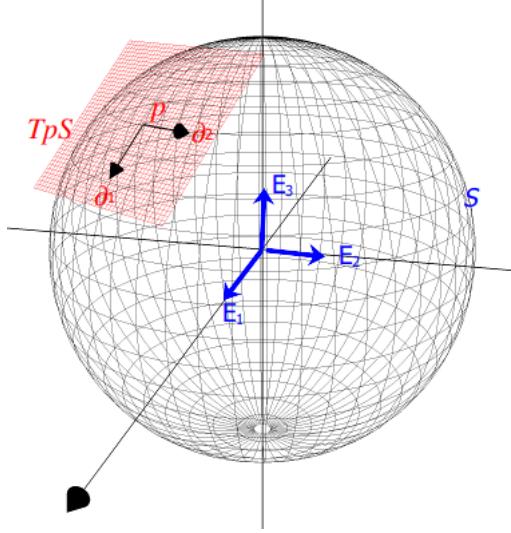
- b. Calculate the normal vector field  $N$  for the ENNEPER surface.  
c. Calculate the exterior surface normal  $N_p$  at  $p = (1, 1)$ .  
d. Try the parametrization  $\Psi$  which is given in WOLFRAM [50] to solve b. and c.



We should now have a solid feeling  
for what a normal vector (field) on a surface is.



### 3 The First Fundamental Form



Visualization of the tangential plane  $T_p S$  at  $p$  on surface  $S$   
 Figure 9: with tangential basis vectors  $\partial_1$  and  $\partial_2$  and ambient space  
 basis  $E_1, E_2, E_3$  in  $\mathbb{R}^3$ , cf. REJBRAND [26].

To develop general methods to study properties of surfaces in ambient space, we describe the geometry of a surface by means of a parametrization  $X: U \rightarrow \mathbb{R}^3$  and measurements, which are defined on the parameter set  $U$ . The so-called *metric tensor  $g$  alias the first fundamental form  $I$*  is such a measurement of a surface parametrization  $X$ , that plays an crucial role in Differential Geometry and General Relativity. It allows to define and calculate distances and angles and other measurements on surfaces ('manifolds') in space, just as the inner product do it on a Euclidean space. But be warned: in LINDNER [16, p.33 ff] resp. GRINFELD [11, p.56 ff] we defined and calculated the metric tensor  $\hat{g}$  for the ambient space  $\mathbb{R}^3$  using a coordinate basis in  $\mathbb{R}^3$ , but now we define *a metric in a tangential space  $T_p S$*  of a parametric surface  $S$ ! Looking at fig. 8 we will therefore use *tangential basic vectors  $\partial_i$*  in  $T_p S$  to construct the metric tensor of the parametrization.

**Remark. 1.** The components of the surface metric tensor  $g$  in a tangential space basis  $\partial_1, \partial_2$  is a symmetric matrix (tensor), whose entries transform 'covariantly' under changes to the coordinate system. Thus a surface metric tensor is a covariant symmetric tensor of type  $(0, 2)$ .

2. The surface metric tensor  $g$  is heavily used to do so-called *index juggling*, cf. [11, p.88], i.e. tensor operations of *raising* or *lowering* an index, see [16], [2, p.335 ff], [23, p.25 ff]. For the definition of  $g$  only the tangential space basic of  $X$  will be necessary, the surface normal vector  $N$  will occur crucial in the definition of the *second fundamental form  $II$*  alias the  $2^{nd}$  fundamental tensor  $h$  in the next chapter.

### 3.1 Definition

Let  $X: U \rightarrow \mathbb{R}^3$  be a parametrization of an 2-dimensional parametric surface  $S$ .

Let's denote the parameters  $(u, v)$  for the moment 'tensorial' by  $(u^1, u^2)$  and the basis tangential vectors of  $T_p S$  by

$$\partial = (\partial_1, \partial_2) := (X_{u^1}, X_{u^2}) \equiv \left( \frac{\partial X}{\partial u^1}, \frac{\partial X}{\partial u^2} \right) \quad (3.1)$$

Then the (covariant) *surface metric tensor*  $g = (g_{ij})$  alias the *first fundamental form I* "fff" is defined as the matrix of the pairwise dot products of these basis tangential vectors, i.e.

$$g_{ij} := \partial_i \bullet \partial_j \quad (3.2)$$

#### Remark.

1. The metric tensor is also called GRAM's matrix, therefore the notion  $g_{ij}$ .
2. We have the explicit calculation formula using the parameterization  $X$  of surface  $S$

$$g_{ij} = \frac{\partial X}{\partial u^i} \bullet \frac{\partial X}{\partial u^j} \quad (3.3)$$

3. The following lexicon helps to memorize these conventions.

LEXICON	<i>Math</i>	MAXIMA
covariant surface metric tensor $g$ :	$g_{ij}$	gij or g
contravariant surface metric tensor $g^{-1}$ :	$g^{ij}$	gu or invert(g)

In MAXIMA we sometimes try to mimic the tensor notion of  ${}_{ij}$  resp.  ${}^{ij}$  resp.  ${}^i_j$  by the identifier **dd** (read: 'down down') resp. **uu** (read: 'up up') resp. **ud** (read: 'up down').<sup>6</sup>

4. The metric components of  $g$  are mostly denoted in classic GAUSS writing as

$$E := g_{11} = g(\partial_1, \partial_1)$$

$$F := g_{12} = g(\partial_1, \partial_2)$$

$$G := g_{22} = g(\partial_2, \partial_2)$$

Ergo we have  $g = \begin{bmatrix} E & F \\ F & G \end{bmatrix}$  and  $g^{-1} = \frac{1}{E \cdot G - F^2} \cdot \begin{bmatrix} G & -F \\ -F & E \end{bmatrix}$ .

Cf. e.g. [24, p.98].

5. We prefer the notation *surface metric tensor* for  $g$ . But if one thinks metaphorically at calculating small distances 'ds' between 'two infinitesimally close' points<sup>7</sup> on the surface resp. the tangential plane, one gets the following expression for the points distance

$$I \equiv ds^2 = Edu^2 + 2Fdudv + Gdv^2 = E \cdot du \wedge du + 2F \cdot du \wedge dv + GE \cdot dv \wedge dv \quad (3.4)$$

which is the bilinear noted variant of  $g$ . This motivates the name choice '*1<sup>st</sup> fundamental form*' *I*. Using such differential forms to calculate line length, areas or volumes using CAS EIGENMATH is demonstrated in [16]. Therefore we will not discuss measurements like distances or volumes here. Nevertheless you find an example here [8, p.14 ff] or [20, p.172].

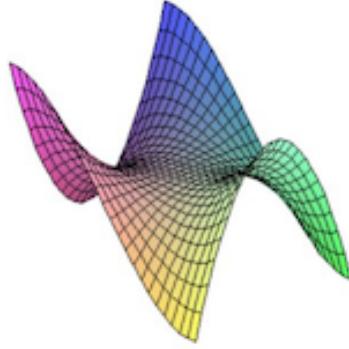
---

<sup>6</sup>This convention was introduced by G. WEIGT in his script about the SCHWARZSCHILD metric, cf. [45].

<sup>7</sup>More about this route e.g. in [4, p.170 ff] or [24, p.98].

### 3.2 Implementation and Examples

**Example 8.** (The surface metric tensor  $g$  for the monkey saddle) One of the most elementary examples is the 2-dimensional monkey saddle, cf. RECKZIEGEL [25, p.103],



or google for [▷wolfram monkey saddle](#).

A parametrization for the monkey saddle is

$$\begin{aligned} X: U &\rightarrow \mathbb{R}^3 \\ (u, v) &\mapsto (u, v, u^3 - 3uv^2) \end{aligned}$$

We use MAXIMA to calculate the surface metric tensor  $g$  and the 1<sup>st</sup> fundamental form  $I$ .

```
/* Maxima - DiffGeo06.txt */
"Monkey saddle";
define( X(u,v) , [u,v, u^3 - 3*u*v^2]);

Xu : diff(X(u,v),u);           /* partial derivative w.r.t. u */
Xv : diff(X(u,v),v);

E : Xu.Xu;                     /* dot product of basis vectors */
F : Xu.Xv;
G : Xv.Xv;

g : matrix( [E,F], [F,G] );   /* fff in matrix shape */
expand(g);

guu : invert(g);              /* the contravariant fff */
expand(guu);

g: matrix( [0,0], [0,0]);     /* alternativ construction */
g[1,1] : Xu.Xu;
g[1,2] : Xu.Xv;
g[2,1] : g[1,2];
g[2,2] : Xv.Xv;

expand(g);
```

```
/* fff I alias ds^2 in differential form (line element) */
I : E*dudu + 2*F*dudv + G*dvdv      ;
expand(I);
```

▷ To open MAXIMA *on line* [click here](#).

MAXIMA output:

```
(%o14) (3 u2-3 v2)2+1
(%o15) -6 u v (3 u2-3 v2)
(%o16) -6 u v (3 u2-3 v2)
(%o17) 36 u2 v2+1
(%o18) 
$$\begin{pmatrix} 9 v^4-18 u^2 v^2+9 u^4+1 & 18 u v^3-18 u^3 v \\ 18 u v^3-18 u^3 v & 36 u^2 v^2+1 \end{pmatrix}$$

(ds2) dudu ((3 u2-3 v2)2+1)+dvdv (36 u2 v2+1)-12 dudv u v (3 u2-3 v2)
(%o20) 9 dudu v4+36 dudv u v3+36 dvdv u2 v2-18 dudu u2 v2-36 dudv u3 v+9 dudu u4+dvdv+dudu
```

We see  $E, F, G$  in %o14..%16 and  $g$  displayed expanded in %o18 and in classic differential form in (ds2).

### Exercise 28.

- a. Reproduce the plot of the monkey saddle with CALC PLOT3D or MAXIMA.
- b. A textbook gives the first fundamental form " $I$ " of the monkey saddle by

$$I = [1 + (3 u^2 - 3 v^2)^2] du^2 - 2 [18 u v (u^2 - v^2)] du dv + (1 + 36 u^2 v^2) dv^2$$

How to reconstruct the coefficients  $E, F, G$  from  $I$ ?

### 3.3 Exercises

**Exercise 29.** Calculate the surface metric tensor  $g$  for the 'saddle' surface  $z = xy$ .  
*Hint:* parametrization  $X$  by  $X(u, v) = (u, v, u \cdot v)$ .

**Exercise 30.** Let  $S$  denote the surface parametrized by  $X(u, v) = (u, v, u^2 + v^2)$ ,  $u, v \in \mathbb{R}$ . Calculate the surface metric tensor  $g$  and the fff  $I$  for this surface.

**Exercise 31.** (The surface metric tensor  $g$  for a *function surface*)

A parameterization of the graph of  $f$  with  $z = f(x, y)$  is  $X(u, v) = (u, v, f(u, v))$ .

a. Verify with MAXIMA:

$$E = 1 + f_u^2 \quad F = f_u \cdot f_v \quad G = 1 + f_v^2$$

b. Calculate the surface metric tensor  $g$  and the fff  $I$  for the function graph.

**Exercise 32.** (The surface metric tensor  $g$  for the *elliptic paraboloid*)

The *elliptic paraboloid* is the parametric surface defined by  $X(u, v) = (u, v, u^2 + v^2)$ .

- a. Verify the coefficients of  $g$  using MAXIMA:  $E = 1 + 4u^2$ ,  $F = 4uv$ ,  $G = 1 + 4v^2$ .  
 b. Then write down the fff as  $I = ds^2 = \dots$

- You may cross-check your results of the following exercises in [?].

- Calculate the coefficients of the surface metric tensor  $g = \begin{bmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{bmatrix} = \begin{bmatrix} E & F \\ F & G \end{bmatrix}$  alias the 1<sup>st</sup> fundamental form  $I$  using MAXIMA for the following surfaces ...

**Exercise 33.** the cylinder  $X(u, v) = (a \cos(v), a \sin(v), u)$ . Choose  $a = 2$ .

**Exercise 34.** the cone  $X(u, v) = (u \cos(v), u \sin(v), u)$ .

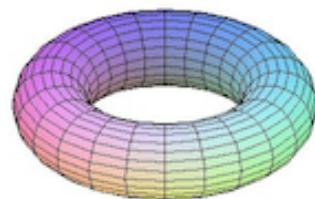
**Exercise 35.** hyperbolic paraboloid  $X(u, v) = (u, v, u^2 - v^2)$ .

**Exercise 36.** the helicoid  $X(u, v) = (v \cos(\alpha u), v \sin(\alpha u), u)$ . Choose  $\alpha = 1$ .

**Exercise 37.** the unit sphere  $X(u, v) = (\cos(u) \sin(v), \sin(u) \sin(v), \cos(v))$ .

**Exercise 38.** the catenoid  $X(u, v) = (c \cosh \frac{v}{c} \cos u, c \cosh \frac{v}{c} \sin u, v)$ . Set  $c = 1$ .

**Exercise 39.** the torus  $X(u, v) = ((b + a \cos(v)) \cos(u), (b + a \cos(v)) \sin(u), a \sin(v))$ .



Verify:  $E = (b + a \cos(v))^2$ ,  $F = 0$ ,  $G = a^2$ .

**Exercise 40.** the PLÜCKER Conoid  $X(u, v) = (v \cdot \cos(u), v \cdot \sin(u), 2 \cos(u) \cdot \sin(u))$ .

**Exercise 41.** the SCHERK surface  $X(u, v) = (u, v, \ln(\frac{\cos v}{\cos u}))$ .

**Exercise 42.** the ENNEPER surface  $X(u, v) = (u - \frac{1}{3}u^3 + uv^2, v - \frac{1}{3}v^3 + vu^2, u^2 - v^2)$ .

**Exercise 43.** the BELTRAMI pseudosphere

$$X(u, v) = (\sin u \cos v, \sin u \sin v, \cos u + \ln(\sin u(\cos u + 1))).$$

**Exercise 44.** the hexenhet  $X(u, v) = (\frac{\alpha}{\sqrt{u}} \cdot \cos v, \frac{\alpha}{\sqrt{u}} \cdot \sin v, u)$  where  $\alpha^2 = \frac{2}{3\sqrt{3}}$ .

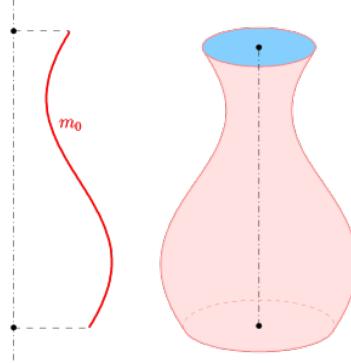
**Exercise 45.** the BIANCHI surface

$$X(u, v) = \begin{bmatrix} \frac{2 \cdot \sqrt{v^2+1} \cdot \sin(u) \cdot \cos(-v+\arctan(v))}{1+v^2 \sin(u)^2} \\ \frac{-2 \cdot \sqrt{v^2+1} \cdot \sin(u) \cdot \sin(-v+\arctan(v))}{1+v^2 \sin(u)^2} \\ \ln(\tan(\frac{1}{2u})) + \frac{2 \cdot \cos(u)}{1+v^2 \sin(u)^2} \end{bmatrix} \quad (3.5)$$

**Remark.** Check your results e.g. by visiting  $\triangleright$  REJBRAND's Encyclopedia.

**Exercise 46.** (The surface metric tensor  $g$  for a *surface of revolution*)

A *surface of revolution* is obtained by rotating a graph of  $f: ]a, b[ \rightarrow \mathbb{R}$  around an coordinate axis.



A parameterization is given by

$$X(u, v) = (f(u) \cos(v), f(u) \sin(v), g(u)) \quad (3.6)$$

Calculate the surface metric tensor  $g$  for a surface of revolution parametrized by (3.6).

**Remark.** For more information see e.g.  $\triangleright$  Wolfram: SurfaceOfRevolution, [2, p.113], [15, p.218], [24, p.80], [25, p.77],

**Exercise 47.** (The surface metric tensor  $g$  for a *ruled surface*)

A *ruled surface* is a surface  $S$  that can be swept out by moving a line in space, i.e it is a surface which is a collection of straight lines, called the *rulings* of the surface:

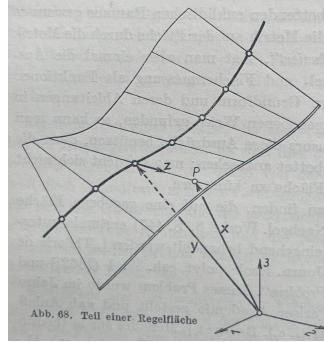


Abb. 68. Teil einer Regelfläche KREYSZIG, p. 218

Let  $c: [a, b] \rightarrow \mathbb{R}^3$  be a parametrization of a curve with  $c(u) \in S$ , i.e. each point of  $c$  meets a ruling. Let  $E(u) \in \mathbb{R}^3$  be a non-zero unit vector positioned at  $c(u)$ .

Then each point of the surface  $S$  is given by the parametrization

$$X(u, v) = c(u) + v \cdot E(u) \quad (3.7)$$

- E.g. PLÜCKER's Conoid can be written as a ruled surface via the decomposition

$$X(u, v) = \begin{bmatrix} v \cdot \cos(u) \\ v \cdot \sin(u) \\ 2 \cos(u) \cdot \sin(u) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 2 \cos(u) \cdot \sin(u) \end{bmatrix} + v \cdot \begin{bmatrix} \cos(u) \\ \sin(u) \\ 0 \end{bmatrix} \quad (3.8)$$

**Remark.** ▷ wolfram: Ruled Surface

**Exercise 48.** (*ruled surfaces vs. surfaces of revolution*)

Some of the classical surfaces studied above are constructible as a surface of revolution or as ruled surface. Fill in and complete the following table:

LEXICON surface is ...	ruled with $c$ and $\vec{r}$	revolution with $f$ and $g$
$c = (0, 0, 2 \cos u \sin u)$	$E = (\cos u, \sin u, 0)$ : PLÜCKER's Conoid	cone
	$f(u) = g(u) = u$ :	cylinder
$f(u) = r = \text{const}$	$g(u) = u$ :	unit sphere
$f = \cos u$	$g = \sin u$ :	...
...	...	...

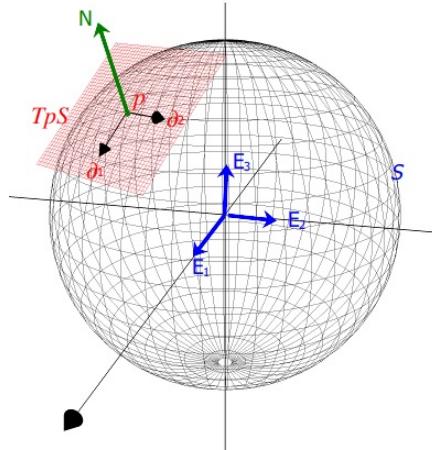
with  $\dots \in \{\text{helicoid, catenoid, hyp. paraboloid, torus, SCHERK, ENNEPER, pseudosphere, ..}\}$ .

- Calculate the surface metric tensor  $g$  for a general defined ruled surface by (3.7).
- Verify the results of the calculated values in exercise 33 – 45 using the general formula.

**Remark.** ▷ wiki: Ruled surface.

## 4 The Second Fundamental Form

Once again, now with  $N$ :



Visualization of the tangential plane  $T_pS$  at  $p$  on surface  $S$

Figure 10: with tangential basis vectors  $\partial_1$  and  $\partial_2$ , ambient space basis  $E_1, E_2, E_3$  and **normal vector  $N$**  in  $\mathbb{R}^3$ , cf. REJBRAND [26].

To develop general methods to study properties and the shape of surfaces in ambient space, we describe the geometry of a surface by means of a parametrization  $X: U \rightarrow \mathbb{R}^3$  and measurements, which are defined on the parameter set  $U$ . The metric tensor  $g$  alias the **fff  $I$**  allowed to measure distances, angles and areas on  $S$ .

To measure how 'curved' a surface is, we introduced the GAUSS map resp. the surface normal vector field  $N$  and got a first feeling of the curvature of  $S$  by watching the moving  $N$  on the surface. To do this, we have to *consider the surface from the outside* by means of the **exterior** normal  $N$ , i.e. look at  $S$  as an object of the ambient space  $\mathbb{R}^3$ , see fig.10.

Various terms have emerged to cope with the curvature of  $S$ , each of which sheds light on different aspects: the GAUSSIAN curvature, the mean curvature, the normal curvature, the principal curvatures etc. This seems quite confusing at first, but fortunately all these characteristics can ultimately be traced back to the so-called *shape operator  $A$*  and the so-called  *$2^{nd}$  fundamental tensor  $h$*  alias the *sff  $II$* .

So the precise tools to measure the curvature are the **sff  $h$**  and the **shape operator  $A$** , which will be defined and implemented first. For the definition of the surface metric ,i.e. the **fff  $g$** , there was only the tangential space basis of the parameterization  $X$  necessary, whereas the surface normal vector  $N$  is now crucial in the construction of the  *$2^{nd}$  fundamental form  $h$*  and the **shape operator  $A$** .

All this will be studied in this chapter.

## 4.1 Definition

Let  $X: U \rightarrow \mathbb{R}^3$  be a parametrization of an 2-dimensional parametric surface  $S$ .

Let  $X_u$  and  $X_v$  be the canonical basis in the tangential space  $T_p S$ .

Let  $g = (g_{ij})$  be its surface metric tensor and  $V := \sqrt{\det(g)}$  ist volume element.

Let  $\vec{n} := \frac{X_u \times X_v}{|X_u \times X_v|}$  be the standard unit normal vector field on  $T_p S$ .

Let's denote the parameters  $(u, v)$  for the moment tensorial by  $(u^1, u^2)$ .

We then define the coefficients of the *2<sup>nd</sup> fundamental tensor  $h$*  by

$$h_{ik} := X_{u^i u^k} \bullet \vec{n} \quad (4.1)$$

or more explicit

$$h_{ik} := \frac{\partial^2 X}{\partial u^i \partial u^k} \bullet \vec{n} = \frac{1}{V} \cdot \det\left(\frac{\partial X}{\partial u^1}, \frac{\partial X}{\partial u^2}, \frac{\partial^2 X}{\partial u^i \partial u^k}\right) \quad (4.2)$$

### Remark.

1. Equation (4.1) is in PRESSLEY [24, p.124], equation (4.2) in RECKZIEGEL [25, p.110].
2. The components of the 2<sup>nd</sup> fundamental tensor  $h$  are denoted after GAUSS by

$$\begin{aligned} \textcolor{blue}{L} &:= h_{11} = X_{uu} \bullet \vec{n} \\ \textcolor{blue}{M} &:= h_{12} = X_{uv} \bullet \vec{n} \\ \textcolor{blue}{N} &:= h_{22} = X_{vv} \bullet \vec{n} \end{aligned}$$

Ergo we have  $h = \begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{bmatrix} = \begin{bmatrix} L & M \\ M & N \end{bmatrix} = \begin{bmatrix} X_{uu} \bullet \vec{n} & X_{uv} \bullet \vec{n} \\ X_{uv} \bullet \vec{n} & X_{vv} \bullet \vec{n} \end{bmatrix}$ .

The following lexicon helps to identify these coefficients  $L, M, N$  in other textbooks, which use sometimes other letters:

LEXICON	<i>Math</i>	<i>MAXIMA</i>
GAUSS:	$L, M, N$	$L, M, N$
DINEEN [7, 201], SHIFRIN [33, p.45 ff]:	$l, m, n$	$l, m, n$
PRESSLEY [24, 124] or [23, p.94] or [20, p.175] or [15, p.146]:	$L, M, N$	$L, M, N$
BANCHOFF [2, 180], WOLFRAM <i>mathworld</i> :	$e, f, g$	$e, f, g$

Using MAXIMA we prefer mostly  $L, M, N$  and therefore denote the normal vector in a mathematical context by  $\vec{n}$  and in MAXIMA scripts by  $\mathbf{n}$  to avoid identifier collision with the coefficient  $N$  of the 2<sup>nd</sup> fundamental tensor  $h$ .

3. One calls the expression

$$II := L du^2 + 2M dudv + N dv^2 \equiv L \cdot du \wedge du + 2M \cdot du \wedge dv + N \cdot dv \wedge dv$$

the *second fundamental form II* (abbreviated *sff*) of  $X$ .

4. For more information on  $h = II$  see for example [24, p.98], [2, p.178 ff] or [20, p.171].

## 4.2 Implementation and Examples

These constructions should become clear by studying some examples with MAXIMA.

**Example 9.** (the second fundamental tensor of the 'saddle' surface  $z = xy$ )

We use MAXIMA to calculate the second fundamental tensor  $h$  and the second fundamental form  $II$  for the surface  $S$ :  $z = xy$ , cf. [7, p.202].

The parametrization for the function surface  $S$  is  $X(u, v) = (u, v, u \cdot v)$ .

Calculate the coefficients  $L, M, N$  of the 2<sup>nd</sup> fundamental tensor  $h$ . Determine  $I$ .

*Solution.* We use MAXIMA.

1<sup>st</sup> we use RHS of equation (4.2) and use the well known identity of linear algebra  $\det(X_1, X_2, X_{ik}) = (X_1 \times X_2) \bullet X_{ik}$  in (1), wherein  $X_1 \times X_2$  is the normal vector (not yet normed to length 1):

```
/* MAXIMA - DiffGeo07.txt */
"saddle";
define( X(u,v) , [u,v, u*v]);

Xu : diff(X(u,v),u);      /* partial derivative w.r.t. u */
Xv : diff(X(u,v),v);

g: matrix( [0,0], [0,0]);
  g[1,1] : Xu.Xu;
  g[1,2] : Xu.Xv;
  g[2,1] : g[1,2];
  g[2,2] : Xv.Xv;
expand(g);

V : sqrt(determinant(g));

define( cross(v,w), [v[2]*w[3] - w[2]*v[3],
                    w[1]*v[3] - v[1]*w[3],
                    v[1]*w[2] - w[1]*v[2]]);

Xuu : diff(X(u,v),u,2);    /* 2nd partial derivative w.r.t. u */
Xuv : diff( diff(X(u,v),u), v);
Xvv : diff(X(u,v),v,2);

L : 1/V * cross(Xu, Xv) . Xuu; /* (1) */
M : 1/V * cross(Xu, Xv) . Xuv;
N : 1/V * cross(Xu, Xv) . Xvv;

h : matrix( [L,M], [M,N] ); /* sff in matrix shape */
expand(h);
```

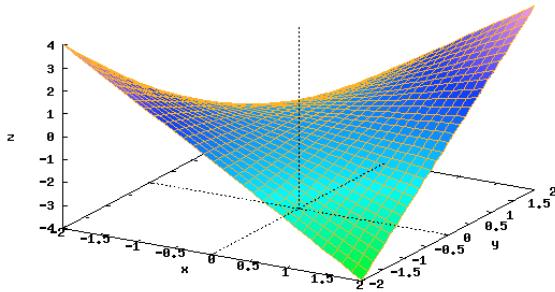
- ▷ Open MAXIMA<sup>on line</sup> or ▷ SAGEMATHCELL

MAXIMA output:

(L)	0
(M)	$\frac{1}{\sqrt{(u^2+1)(v^2+1)-u^2 v^2}}$
(N)	0

and

```
wxplot3d([u,v, u*v], [u,-2,2], [v,-2,2])$
```



**2<sup>nd</sup>** We use LHS of equation (4.2) resp. equation (4.1) and to avoid the  $\det(\cdot)$  – which incorporate implicit the normal vector  $\vec{n}$  using the cross product – we now have to calculate explicit the normal vector  $\vec{n}$  and the corresponding unit normal  $\text{No}$ :

```
/* MAXIMA - DiffGeo08.txt */
"saddle";
define( X(u,v) , [u,v, u*v]);

Xu : diff(X(u,v),u);      /* partial derivative w.r.t. u */
Xv : diff(X(u,v),v);

Xuu : diff(X(u,v),u,2);    /* 2nd partial derivative w.r.t. u */
Xuv : diff( diff(X(u,v),u) , v );
Xvv : diff(X(u,v),v,2);

define( cross(v,w),
        [v[2]*w[3]-w[2]*v[3] , w[1]*v[3]-v[1]*w[3] , v[1]*w[2]-w[1]*v[2]]);

mag(v) := sqrt(v.v);

"unit normal vector";
n      : cross(Xu,Xv);
magN   : mag(n);
```

```
No : n/magN;

L : Xuu . No; /* (1) */
M : Xuv . No;
N : Xvv . No;

"sff II in matrix shape using (4.1)";
h : matrix( [L,M], [M,N] );
```

- ▷ Open MAXIMA<sup>on line</sup> or ▷ SAGEMATHCELL

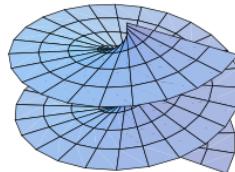
MAXIMA output:

$$(h) \begin{pmatrix} 0 & \frac{1}{\sqrt{v^2+u^2+1}} \\ \frac{1}{\sqrt{v^2+u^2+1}} & 0 \end{pmatrix}$$

It follows:  $II = 2 \cdot \frac{1}{\sqrt{u^2+v^2+1}} dudv$ .

**Example 10.** (the second fundamental tensor  $h$  of the helicoid)

The *helicoid* is the 2D parametric surface defined by  $X(u, v) = (v \cos(u), v \sin(u), u)$ .



Calculate the coefficients  $l, m, n$  of the 2<sup>nd</sup> fundamental tensor  $h$ .

*Solution.* We use wxMAXIMA. We use equation (4.1). Therefore we have to calculate  $\vec{n}$  and  $h$ . We name the coefficients of the 2<sup>nd</sup> fundamental form  $l, m, n$  to avoid name clash with the normal vector  $\vec{n} = N$ .

```
/* MAXIMA - DiffGeo09.txt */
"helicoid";
define( X(u,v) , [v*cos(u),v*sin(u), u] );

Xu : diff(X(u,v),u);          /* partial derivative w.r.t. u */
Xv : diff(X(u,v),v);

Xuu : diff(X(u,v),u,2);      /* 2nd partial derivative w.r.t. u */
Xuv : diff( diff(X(u,v),u) , v);
Xvv : diff(X(u,v),v,2);
```

```

define( cross(v,w),
        [v[2]*w[3]-w[2]*v[3], w[1]*v[3]-v[1]*w[3], v[1]*w[2]-w[1]*v[2]]);

mag(v) := sqrt(v.v);

"unit normal vector";
N      : cross(Xu,Xv);
N      : trigsimp(N);          /* (1) */
magN   : trigsimp(mag(N));    /* (2) */
No : N/magN;

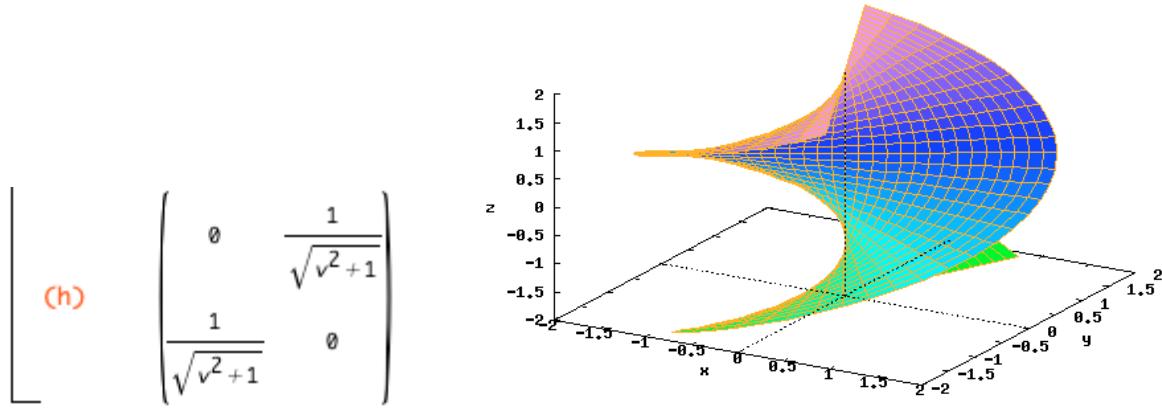
"2nd fundamental form";
l : Xu . No;
m : Xuv . No;
m : trigsimp(radcan(m));     /* (3) */
n : Xvv . No;

"sff h in matrix shape using (4.1)";
h : matrix( [l,m], [m,n] );

wxplot3d([v*cos(u),v*sin(u), u], [u,-2,2], [v,-2,2])$
```

▷ Open MAXIMA<sup>on line</sup> ▷ SAGEMATHCELL. ❤: the plot command is only working in **wxMAXIMA**!

MAXIMA output:



**Remark.** In (1) and (2) we reduce the expression with **trigsimp**, which uses the trigonometric equation  $c^2 + s^2 = 1$ . In (3) we first simplify the fraction by means of **radcan** and afterwards we trig-simplify the intermediate result.

### 4.3 Exercises

- You may cross-check your results of the following exercises in [18].

**Exercise 49.** Calculate the 2<sup>nd</sup> fundamental tensor  $h$  for the unit sphere  $S$ .

Look for ▷ WOLFRAM: sphere and use their parameterization.

Check your result with that given by WOLFRAM.

You may switch inside this page to WOLFRAM|alpha.

**Exercise 50.** Calculate the 2<sup>nd</sup> fundamental tensor  $h$  for the catenoid  $S$ .

Look for ▷ WOLFRAM: catenoid and use their parameterization.

Check your result with that given by WOLFRAM.

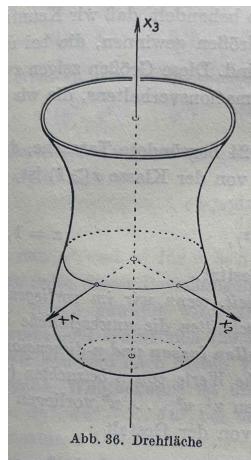
**Exercise 51.** Calculate the 2<sup>nd</sup> fundamental tensor  $h$  for the torus  $S$ .

Look for ▷ WOLFRAM: torus and use their parameterization with  $a = 1$ .

Check your result with that given by WOLFRAM.

**Exercise 52.** (The 2<sup>nd</sup> fundamental tensor  $h$  for a *surface of revolution*)

The surface is obtained by rotating the graph of  $f: ]a, b[ \rightarrow \mathbb{R}$  around an axis.



taken from KREYSZIG, p.107

The parameterization is known to be  $X(u, v) = (f(u) \cos(v), f(u) \sin(v), g(u))$ . Calculate the 2<sup>nd</sup> fundamental tensor  $h$  for a surface of revolution.

**Exercise 53.** (The 2<sup>nd</sup> fundamental tensor  $h$  for a *function surface*)

Let  $z = f(x, y)$  be a function surface.

A parameterization of the graph of  $f$  is  $X(u, v) = (u, v, f(u, v))$ .

a. Verify with MAXIMA the coefficients of  $h$ :

$$\begin{aligned} L &= (1 + f_u^2 + f_v^2)^{-2} \cdot f_{uu} \\ M &= (1 + f_u^2 + f_v^2)^{-2} \cdot f_{uv} \\ N &= (1 + f_u^2 + f_v^2)^{-2} \cdot f_{vv} \end{aligned}$$

b. Give  $h$  in matrix shape and write down the sff  $\Pi$  for the function graph.

**Exercise 54.** (The 2<sup>nd</sup> fundamental tensor  $h$  for the *monkey saddle*).

A parametrization for the monkey saddle is  $X: U \rightarrow \mathbb{R}^3$  by  $(u, v) \mapsto (u, v, u^3 - 3uv^2)$ .

a. Give the second fundamental tensor  $h$  in matrix shape  $(h_{ij})$ .

b. Write down the sff in differential form  $II = \dots$

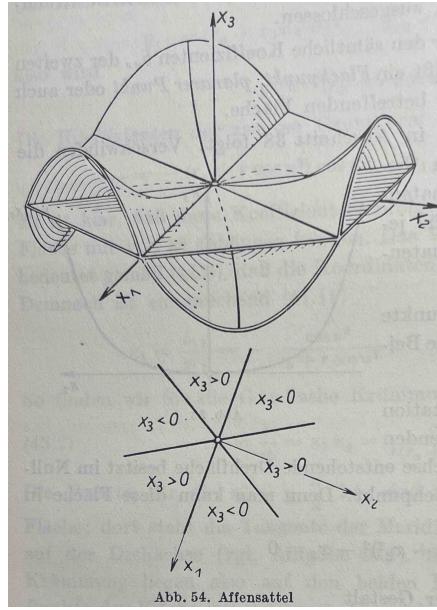


Abb. 54. Affensattel

taken from KREYSZIG, p.168

- You may cross-check your results of the following exercises in [?].
- Calculate the coefficients of the 2<sup>nd</sup> fundamental tensor  $h = \begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{bmatrix} = \begin{bmatrix} L & M \\ M & N \end{bmatrix}$  alias the 2<sup>nd</sup> fundamental form  $II$  using MAXIMA for the following 2D surfaces ...

**Exercise 55.** the cylinder  $X(u, v) = (a \cos(v), a \sin(v), u)$ . Choose  $a = 1$ .

**Exercise 56.** the cone  $X(u, v) = (u \cos(v), u \sin(v), u)$ .

**Exercise 57.** the elliptic paraboloid  $X(u, v) = (u, v, u^2 + v^2)$ .

**Exercise 58.** the hyperbolic paraboloid  $X(u, v) = (u, v, u^2 - v^2)$ .

**Exercise 59.** the PLÜCKER conoid  $X(u, v) = (v \cdot \cos(u), v \cdot \sin(u), 2 \cos(u) \cdot \sin(u))$ .

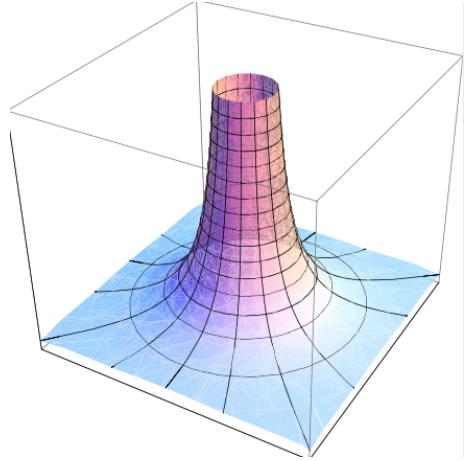
**Exercise 60.** the SCHERK surface  $X(u, v) = (u, v, \ln(\frac{\cos v}{\cos u}))$ .

**Exercise 61.** the ENNEPER surface  $X(u, v) = (u - \frac{1}{3}u^3 + uv^2, v - \frac{1}{3}v^3 + vu^2, u^2 - v^2)$ .  
See [8, p.22].

**Exercise 62.** the BELTRAMI pseudosphere

$$X(u, v) = (\sin u \cos v, \sin u \sin v, \cos u + \ln(\sin u(\cos u + 1))).$$

**Exercise 63.** the hexen hut  $X(u, v) = (\alpha/\sqrt{u} \cdot \cos v, \alpha/\sqrt{u} \cdot \sin v, u)$  where  $\alpha^2 = \frac{2}{3\sqrt{3}}$ , see WHEELER [46, p.21 ff].



**Exercise 64.** the BIANCHI surface

$$X(u, v) = \begin{bmatrix} \frac{2\sqrt{v^2+1}\sin(u)\cos(-v+\arctan(v))}{1+v^2\sin(u)^2} \\ \frac{-2\sqrt{v^2+1}\sin(u)\sin(-v+\arctan(v))}{1+v^2\sin(u)^2} \\ \ln(\tan(\frac{1}{2u})) + \frac{2\cos(u)}{1+v^2\sin(u)^2} \end{bmatrix}$$

**Remark.** Check your results also e.g. by visiting [REJBRAND Encyclopaedia](#).

## 5 Shape Operator and WEINGARTEN Map

We now implement the crucial tool to measure the curvature of a surface parametrized by the parametrization  $X: U \rightarrow \mathbb{R}^3$ , the so-called shape operator  $A$ . An also used name for  $A$  is WEINGARTEN map. To construct  $A$  we need both, the fff  $g$  and the sff  $h$ .

### 5.1 Definition

Let  $X: U \rightarrow \mathbb{R}^3$  be a parametrization of an 2-dimensional parametric surface  $S$ .

Let  $X_u$  and  $X_v$  be the canonical basis in the tangential space  $T_p S$ .

Let  $g = g_{ij}$  be its surface metric tensor and  $g^{-1} = g^{ij}$ , its inverse matrix.

Let  $h = h_{ij}$  be its 2<sup>nd</sup> fundamental tensor.

Then we define the coefficients of the *shape operator*  $A \in \mathbb{R}^{2 \times 2}$

alias the *WEINGARTEN map*  $A$  by

$$A_{ik} := \sum_{j=1}^2 g^{ij} \cdot h_{ik} \quad (5.1)$$

where  $i = 1, 2$  and  $k = 1, 2$ .

**Remark.**

1. Equation (5.1) is from RECKZIEGEL [25, p.137].
2. Using the coefficients  $E, F, G$  of  $g$  and  $L, M, N$  of  $h$  we have explicit

$$\begin{aligned} A &= g^{-1} \bullet h = \begin{bmatrix} E & F \\ F & G \end{bmatrix}^{-1} \bullet \begin{bmatrix} L & M \\ M & N \end{bmatrix} \\ &= \frac{-1}{E \cdot G - F^2} \cdot \begin{bmatrix} G & -F \\ -F & E \end{bmatrix} \bullet \begin{bmatrix} L & M \\ M & N \end{bmatrix} \end{aligned} \quad (5.2)$$

i.e.

$$A = \frac{1}{EG - F^2} \cdot \begin{bmatrix} GL - FM & GM - FN \\ EM - FL & EN - FM \end{bmatrix} \quad (5.3)$$

cf. OLOFF [23, p.108].

3. For more information about the shape operator see e.g. BANCHOFF [2, p.186], THORPE [37, p.55], PRESSLEY [24, p.139], OLOFF [23, p.104 ff] or SHIFRIN [33, p.45 ff].

## 5.2 Examples

**Example 11.** (the shape operator of the saddle surface  $z = xy$ )

A parametrization for this function surface  $S$  is  $X: U \rightarrow \mathbb{R}^3$  with  $(u, v) \mapsto (u, v, u \cdot v)$ . Use MAXIMA to calculate the shape operator  $A$  for the surface  $S: z = xy$ .

*Solution.* We use equation (5.2) and calculate at first  $g$  and  $h$ , then  $A$ .

```
/* MAXIMA - DiffGeo10.txt */
"saddle";
define( X(u,v) , [u,v, u*v]);

"calculate 1st ff alias the metric tensor g";
Xu : diff(X(u,v),u);      /* partial derivative w.r.t. u */
Xv : diff(X(u,v),v);

E : Xu.Xu;                  /* dot product of basis vectors */
F : Xu.Xv;
G : Xv.Xv;

g : matrix( [E,F], [F,G] ); /* fff in matrix shape */

gu : invert(g);            /* the inverse of the metric */
radcan(gu);

"calculate 2nd ff";
Xuu : diff(X(u,v),u,2);    /* 2nd partial derivative w.r.t. u */
Xuv : diff( diff(X(u,v),u) , v );
Xvv : diff(X(u,v),v,2);

V : sqrt(determinant(g));
V : radcan(V);

define( cross(v,w),
        [v[2]*w[3]-w[2]*v[3], w[1]*v[3]-v[1]*w[3], v[1]*w[2]-w[1]*v[2]]);

L : 1/V * cross(Xu, Xv) . Xuu;
M : 1/V * cross(Xu, Xv) . Xuv;
N : 1/V * cross(Xu, Xv) . Xvv;

h : matrix( [L,M], [M,N] );

"shape operator";
A : gu . h;
A : radcan(A);
```

•  $\triangleright$  MAXIMA<sup>on line</sup>  $\triangleright$  SAGEMATHCELL

MAXIMA output:

$$(A) \begin{pmatrix} -\frac{uv}{(v^2+u^2+1)^{3/2}} & \frac{u^2+1}{(v^2+u^2+1)^{3/2}} \\ \frac{v^2+1}{(v^2+u^2+1)^{3/2}} & -\frac{uv}{(v^2+u^2+1)^{3/2}} \end{pmatrix}$$

**Example 12.** (the shape operator of the helicoid)

The *helicoid* is the 2D parametric surface defined by  $X(u, v) = (v \cos(u), v \sin(u), u)$ .

Use MAXIMA to calculate the shape operator  $A$  for the helicoid.

*Solution.* We use wxMAXIMA. We know from previous examples the coefficients of the first and the second fundamental form of  $X$ .

```
/* MAXIMA - DiffGeo11.txt */
"Helicoid";
"we know the metric tensor g";
E : 1;
F : 0;
G : v^2+1;
g : matrix( [E,F], [F,G] );
gu : invert(g);      /* the inverse of the metric */

"we know the 2nd ff h";
L : 0;
M : 1/sqrt(v^2+1);
N : 0;
h : matrix( [L,M], [M,N] );
"shape operator of helicoid";
A : gu . h;
```

•  $\triangleright$  MAXIMA<sup>on line</sup>  $\triangleright$  SAGEMATHCELL

MAXIMA output:

$$(A) \begin{pmatrix} 0 & \frac{1}{\sqrt{v^2+1}} \\ \frac{1}{(v^2+1)^{3/2}} & 0 \end{pmatrix}$$

**Exercise 65.** (the shape operator of the elliptic paraboloid)

The *elliptic paraboloid* is the parametric surface defined by  $X(u, v) = (u, v, u^2 + v^2)$ .

a. Verify using MAXIMA:  $E = 1 + 4u^2, F = 4uv, G = 1 + 4v^2$ .

b. Verify using MAXIMA:  $L = 2(4u^2 + 4v^2 + 1)^{-1/2}, M = 0, N = L$ .

c. Calculate the shape operator for the elliptic paraboloid along the pattern of example 11 and alternative along example 12.

### 5.3 Exercises

- You may cross-check your results of the following exercises in [18].
- Calculate the shape operator  $A = g^{-1} \bullet h$  using MAXIMA for the following 2D surfaces
- ...

**Exercise 66.** the cylinder  $X(u, v) = (a \cos(v), a \sin(v), u)$ . Choose  $a = 1$ .

**Exercise 67.** the cone  $X(u, v) = (u \cos(v), u \sin(v), u)$ .

**Exercise 68.** the hyperbolic paraboloid  $X(u, v) = (u, v, u^2 - v^2)$ .

**Exercise 69.** the monkey saddle  $X(u, v) = (u, v, u^3 - 3uv^2)$ .

**Exercise 70.** the unit sphere  $X(u, v) = (\cos(u) \sin(v), \sin(u) \sin(v), \cos(v))$ .

**Exercise 71.** the catenoid  $X(u, v) = (c \cosh \frac{v}{c} \cos u, c \cosh \frac{v}{c} \sin u, v)$ .

**Exercise 72.** the torus  $X(u, v) = ((b + a \cos(v)) \cos(u), (b + a \cos(v)) \sin(u), a \sin(v))$ .

**Exercise 73.** the function graph  $X(u, v) = (u, v, f(u, v))$ .

**Exercise 74.** the surface of revolution  $X(u, v) = (f(u) \cos(v), f(u) \sin(v), g(u))$ .

**Exercise 75.** the PLÜCKER conoid  $X(u, v) = (v \cdot \cos(u), v \cdot \sin(u), 2 \cos(u) \cdot \sin(u))$ .

**Exercise 76.** the SCHERK surface  $X(u, v) = (u, v, \ln(\frac{\cos v}{\cos u}))$ .

**Exercise 77.** the ENNEPER surface  $X(u, v) = (u - \frac{1}{3}u^3 + uv^2, v - \frac{1}{3}v^3 + vu^2, u^2 - v^2)$ .

**Exercise 78.** the BELTRAMI pseudosphere

$$X(u, v) = (\sin u \cos v, \sin u \sin v, \cos u + \ln(\sin u(\cos u + 1)))$$

**Exercise 79.** the hexen hut  $X(u, v) = (\alpha/\sqrt{u} \cdot \cos v, \alpha/\sqrt{u} \cdot \sin v, u)$ .

**Exercise 80.** the BIANCHI surface

$$X(u, v) = \begin{bmatrix} \frac{2 \cdot \sqrt{v^2 + 1} \cdot \sin(u) \cdot \cos(-v + \arctan(v))}{1 + v^2 \sin(u)^2} \\ \frac{-2 \cdot \sqrt{v^2 + 1} \cdot \sin(u) \cdot \sin(-v + \arctan(v))}{1 + v^2 \sin(u)^2} \\ \ln(\tan(\frac{1}{2u})) + \frac{2 \cdot \cos(u)}{1 + v^2 \sin(u)^2} \end{bmatrix}$$

**Remark.** One can check some of the results also by visiting  $\triangleright$  REJBRAND Encyclopaedia.

## 6 Scalar Curvature Measures

There are many concepts that have emerged to cope with the curvature of a surface  $S$ , each of which sheds light on different aspects of the idea of curvature: the **GAUSSIAN curvature**, the mean curvature, the normal curvature, the principal curvatures etc. We now define some of these characteristics of a surface  $S$  by means of the *shape operator*  $A$ .

### 6.1 Definition

Let  $X: U \rightarrow \mathbb{R}^3$  be a parametrization of an 2-dimensional parametric surface  $S$ .

Let  $g$  be the surface metric tensor of  $X$  with the coefficients  $E, F, G$ .

Let  $h$  be the  $2^{nd}$  fundamental tensor of  $X$  with the coefficients  $L, M, N$ .

Let  $A$  be the shape operator of  $X$ .

We then define

- a. The **GAUSS curvature**  $K$  is the real-valued function  $K: U \rightarrow \mathbb{R}$  with

$$p \mapsto K(p) := \det(A(p)) = \frac{LN - M^2}{EG - F^2} \quad (6.1)$$

- b. The **mean curvature**  $H$  is the real-valued function  $H: U \rightarrow \mathbb{R}$  with

$$p \mapsto H(p) := \frac{1}{2} \cdot \text{trace}(A(p)) = \frac{EN + GL - 2FM}{2(EG - F^2)} \quad (6.2)$$

- c. The **principal curvatures**  $\kappa_i \in \mathbb{R}$  of  $S$  are the roots of the equation

$$\det(A - \kappa \cdot I) = 0 \quad (6.3)$$

where  $I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \in \mathbb{R}^{2 \times 2}$  is the 2-by-2 identity matrix, i.e the principal curvatures are the eigenvalues of the shape operator  $A$ .

The *eigenvectors* corresponding to the the principal values  $\kappa_i$  are called the *principal vectors* alias **principal curvature directions**, denoted  $\vec{\kappa}$ .

We have the fact

**Theorem.** *The first and second principal curvatures  $\kappa_1$  and  $\kappa_2$  are the real-valued functions  $\kappa_i: U \rightarrow \mathbb{R}$  with*

$$\kappa_1 := H - \sqrt{H^2 - K} \quad (6.4)$$

$$\kappa_2 := H + \sqrt{H^2 - K} \quad (6.5)$$

**Remark.** 1. Equations (6.1), (6.2) and (6.4), (6.5) are from RECKZIEGEL [25, p.115], ad equation (6.3) cf. PRESSLEY [24, p.132]. See also BANCHOFF [2, p.199].

2. For more information about the curvature measures see e.g. THORPE [37, p.86–89], PRESSLEY [24, p.132; p.147 ff], OLOFF [23, p.108 ff] or SHIFRIN [33, p.49].

## 6.2 Examples

**Example 13.** (GAUSSian, mean and principal curvature of the helicoid)

The *helicoid* is the 2D parametric surface defined by  $X(u, v) = (v \cos(u), v \sin(u), u)$ .

1. Use MAXIMA to calculate the GAUSSian curvature, the mean curvature and the principal curvatures for the helicoid.
2. Calculate the GAUSSian and mean curvature at the point  $p = (1, 1)$ .
3. Write GAUSSian  $K$ , mean  $H$  and principal curvatures  $\lambda$  as a function of an arbitrary point  $p = (u, v)$  on the helicoid.
4. Calculate the principal curvature directions  $\vec{\kappa}_i(p)$  at  $p = (1, 1)$ .

*Solution.* This example is a bit longer.

1<sup>st</sup> We use (6.1) and (6.2). We comment after the code lines and the output.

```
/* MAXIMA - DiffGeo12.txt */
"Helicoid";
define( X(u,v) , [v*cos(u),v*sin(u), u]);

"0."; /* prerequisites */

"we know the metric tensor g";
E : 1;
F : 0;
G : v^2+1;
g : matrix( [E,F], [F,G] );
gu : invert(g);

"we know the 2nd ff h";
L : 0;
M : 1/sqrt(v^2+1);
N : 0;
h : matrix( [L,M], [M,N] );

"shape operator of helicoid";
A : gu . h; /* (0) */

"1.";

"GAUSS curvature via (6.1)";
K : determinant(A); /* (1) */

"mean curvature via (6.2)";
trace(M) := M[1,1] + M[2,2]; /* trace of matrix M */
H : trace(A); /* (2) */
```

```

"principal curvature via (6.3)";
E : matrix( [1,0], [0,1] )$ 
S : solve( determinant(A - lambda*E)=0, lambda); /* S olution set (3) */

lambda1 : S[1];                                /* (4) */
lambda2 : S[2];

```

- ▷ Paste the code in MAXIMA *on line* or ▷ SAGEMATHCELL wxMAXIMA output:

```

(%o17) 1.
(%o18) GAUSS curvature via (6.1)
(K)   - $\frac{1}{(v^2+1)^2}$ 
(%o20) mean curvature via (6.2)
(%o21) trace(M):=M1,1+M2,2
(H)   0
(%o23) principal curvature via (6.3)
(S)   [lambda=- $\frac{1}{v^2+1}$ , lambda= $\frac{1}{v^2+1}$ ]
(lambda1) lambda=- $\frac{1}{v^2+1}$ 
(lambda2) lambda= $\frac{1}{v^2+1}$ 

```

**Comment.** We took the coefficients of the fff and the sff from results before. Then the shape operator is calculated in (0) via formula (5.2). In (1) the GAUSS curvature  $K$  is calculated and in (2) the mean curvature  $H$ , whereby the trace of  $A$  is defined by an user-defined ad hoc function . In (3) we determine the values of principal curvatures  $\lambda_i$ , see (4), which we could get easily by mind from the presented equation.

**2<sup>nd</sup>** We use MAXIMA's **subst** function to evaluate  $K$ ,  $H$  and  $\lambda$  at a specific point in  $X(U)$ .

```

/* K and H and lambda at p=(1,1) */
"2.";

Ap : subst([u=1,v=1], A);                                /* (5) */
Kp : subst([u=1,v=1], K);
Hp : subst([u=1,v=1], H);
lambda1p : subst([u=1,v=1], S[1]);
lambda2p : subst([u=1,v=1], S[2]);

```

wxMAXIMA output:

```
(%o28) 2.
(Ap)   ⎡ 0      1
          ⎢   ───
          ⎢   √2
          ⎣
(Kp)   - 1
          ──
          4
(Hp)   0
(lambda1p) lambda=- 1
          ──
          2
(lambda2p) lambda= 1
          ──
          2
```

**Comment.** (5) calculates the shape operator  $A$  at  $p = (1, 1)$  with the values  $u = 1$  and  $v = 1$ , which are substituted in (0). Therefore  $K_p$ ,  $H_p$  and  $\lambda_p$  are returned with its values evaluated at  $p$ .

**3<sup>rd</sup>** We again use (6.1) and (6.2) to generalize.

```
/* K and H and lambda as functions of p=(u,v). */
"3.";
KK(x,y) := subst([u=x,v=y],K);                                /* (6) */
HH(x,y) := subst([u=x,v=y],H);
L1(x,y) := subst([u=x,v=y],lambda1);
L2(x,y) := subst([u=x,v=y],lambda2);

KK(1,1);                                                        /* (7) */
```

MAXIMA output:

```
(%o34) 3.
(%o35) KK(x,y):=subst([u=x,v=y],K)
(%o36) HH(x,y):=subst([u=x,v=y],H)
(%o37) L1(x,y):=subst([u=x,v=y],lambda1)
(%o38) L2(x,y):=subst([u=x,v=y],lambda2)
(%o39) - 1
          ──
          4
```

**Comment.** We generalize what was done in (5) for a special point. Therefore in (6) we write  $p = (x, y)$  undetermined as arbitrary point on the LHS. A special value of the GAUSS curvature  $K_p$  is then realized via a function call  $KK(1, 1)$  with the input  $(1, 1)$  in (7). So we will be able to easily calculate a bundle of values of  $K$ ! Look at the next example. The same is done for  $H_p$  and  $\lambda_p$ , which are now also functions for arbitrary points on  $U$ .

**4<sup>th</sup>** We use the new functions  $K_p$ ,  $H_p$  and  $\lambda_p$  and build-in functions to calculate the eigenvalues and eigenvector at  $p$ . One could alternatively recur on the definition of these concepts and do a small step by mind or with paper & pencil. See the comment below.

```
/* Eigenvalues and Eigenvectors. */
"4.";

Ap;
charpoly(Ap,x);
eigenvalues(Ap);          /* (8) */
eigenvectors(Ap);        /* (9) */
```

MAXIMA output:

```
(%o40) 4.
(%o41)

$$\begin{pmatrix} 0 & \frac{1}{\sqrt{2}} \\ \frac{1}{z^{3/2}} & 0 \end{pmatrix}$$

(%o42)  $x^2 - \frac{1}{4}$ 
(%o43)  $[-\frac{1}{2}, \frac{1}{2}], [1, 1]$ 
(%o44)  $[[[-\frac{1}{2}, \frac{1}{2}], [1, 1]], [[[1, -\frac{1}{\sqrt{2}}]], [[1, \frac{1}{\sqrt{2}}]]]]$ 
```

**Comment.** If we do not want to use the build-in functions `charpol`, `eigenvalues` and `eigenvectors`, we could use the mathematical definitions of these concepts: We know from command (3), that  $\lambda_p(1, 1) = x^2 - \frac{1}{4}$ . We used MAXIMA's `solve` function to calculate the roots of this quadratic equation – easy done in mind. In (3) we used the well known ansatz  $(A - \lambda \cdot E) = O$  for the determination of eigenvectors for known eigenvalues  $\lambda_i$ . We take from the first line of the returned equation matrix the line

$$\frac{y}{\sqrt{2}} - \frac{1}{2}x = 0$$

which gives e.g. the eigenvector

$$\begin{bmatrix} -1 \\ \frac{-1}{\sqrt{2}} \end{bmatrix}$$

for  $\lambda_1 = -1/2$ . This automated in MAXIMA via (8) and (9).

**Fact.** The principal curvatures  $\kappa_i$  are the roots of the equation  $\det(\mathbf{h} - \kappa \cdot \mathbf{g}) = 0$ , i.e.

$$\det(\mathbf{h} - \kappa \cdot \mathbf{g}) = 0 \quad (6.6)$$

i.e.

$$\begin{vmatrix} L - \kappa E & M - \kappa F \\ M - \kappa F & N - \kappa G \end{vmatrix} = 0 \quad (6.7)$$

This is a simplified version of (6.3), which does not use the shape operator  $A$  at all. Cf. Pressley [24, p.132] or Sochi [32, p.121].

**Exercise 81.** Calculate the principal curvatures  $\kappa$  of the helicoid using (6.7).

**Example 14.** (GAUSS and mean curvature of the elliptic paraboloid)

The *elliptic paraboloid* is the parametric surface defined by

$$X(u, v) = (u, v, u^2 + v^2) \quad (6.8)$$

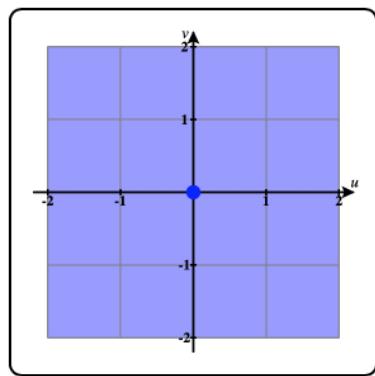
with patch region  $(u, v) \in \mathbb{R}^2$  or by the parametric ansatz defined by

$$X(u, v) = (a\sqrt{u} \cos(v), b\sqrt{u} \sin(v), cu) \quad (6.9)$$

with real constants  $a, b, c$  and patch region  $(u, v) \in \mathbb{R}^+ \times [0, 2\pi]$ , cf. [32, p.14].

For this example we use the parametrization  $X$  in (6.8).

- a. Verify using MAXIMA:  $E = 1 + 4u^2, F = 4uv, G = 1 + 4v^2$ .
- b. Verify using MAXIMA:  $L = 2(4u^2 + 4v^2 + 1)^{-1/2}, M = 0, N = 2(4u^2 + 4v^2 + 1)^{-1/2}$ .
- c. Calculate the GAUSS curvature  $K$  and mean curvature  $H$  for the integer grid points of the square  $[-2, 2] \times [-2, 2]$  in  $\mathbb{R}^2$ . Discuss your observations.



- d. Display a tableau of the values of the curvature  $K$  for 5 different user input points.

*Solution.*

**a:** and **b:** see above. So we have:

```

/* MAXIMA - DiffGeo13.txt */
"elliptic paraboloid";
define( X(u,v) , [u,v, u^2+v^2]);

"We know the values for g and h";
E : 1 + 4*u^2;
F : 4*u*v;
G : 1 + 4*v^2;

L : 2/sqrt(4*u^2+4*v^2+1);
M : 0;
N : 2/sqrt(4*u^2+4*v^2+1);

A : invert(matrix( [E,F] , [F,G] )) . matrix( [L,M] , [M,N] ); /* (0) */
A : radcan(A);

K : determinant(A);
trace(M) := M[1,1]+M[2,2];
H : trace(A);

```

- ▷ Paste the code in MAXIMA *on line* or ▷ SAGEMATHCELL

c: we define functions  $A_p$ ,  $K_p$  and  $H_p$  and prepare the grid in (1):

```

"..., A, K and H as functions of p=(u,v)";

Ap(x,y) := subst([u=x,v=y],A);
Kp(x,y) := subst([u=x,v=y],K);
Hp(x,y) := subst([u=x,v=y],H);

Kp(1,1); /* a tiny test */

Kuv : matrix(                                     /* (1) */
  [ 0,-2,-1,0,1,2],
  [-2, 0, 0,0,0,0],
  [-1, 0, 0,0,0,0],
  [ 0, 0, 0,0,0,0],
  [ 1, 0, 0,0,0,0],
  [ 2, 0, 0,0,0,0]
);

Kuv;

```

wxMAXIMA output:

```

(%o92) Ap(x,y):=subst([u=x,v=y],A)
(%o93) Kp(x,y):=subst([u=x,v=y],K)
(%o94) Hp(x,y):=subst([u=x,v=y],H)
(%o95)  $\frac{4}{81}$ 
(Kuv) 
$$\begin{pmatrix} 0 & -2 & -1 & 0 & 1 & 2 \\ -2 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$


```

**Comment.** We set the two borders of tablau (matrix) as 'axis' in (1), which are intended to be the coordinates, where the curvature values are to be read off, e.g. curvature  $K(-1, 1)$  will be displayed later at position (3, 5) of the matrix tableau.

Now we fill the grid with the corresponding curvature values, so we receive a distribution of  $K_p$ :

```

" ... fill grid";
for i:2 thru 6 do Kuv[2,i] : Kp(-2,i-4)$      /* (3) */
for i:2 thru 6 do Kuv[3,i] : Kp(-1,i-4)$
for i:2 thru 6 do Kuv[4,i] : Kp( 0,i-4)$
for i:2 thru 6 do Kuv[5,i] : Kp( 1,i-4)$
for i:2 thru 6 do Kuv[6,i] : Kp( 2,i-4)$
Kuv;

```

MAXIMA output:

```

(%o103) 
$$\begin{pmatrix} 0 & -2 & -1 & 0 & 1 & 2 \\ -2 & \frac{4}{1089} & \frac{4}{441} & \frac{4}{289} & \frac{4}{441} & \frac{4}{1089} \\ -1 & \frac{4}{441} & \frac{4}{81} & \frac{4}{25} & \frac{4}{81} & \frac{4}{441} \\ 0 & \frac{4}{289} & \frac{4}{25} & 4 & \frac{4}{25} & \frac{4}{289} \\ 1 & \frac{4}{441} & \frac{4}{81} & \frac{4}{25} & \frac{4}{81} & \frac{4}{441} \\ 2 & \frac{4}{1089} & \frac{4}{441} & \frac{4}{289} & \frac{4}{441} & \frac{4}{1089} \end{pmatrix}$$


```

**Comment.** You may read off the curvature value  $K_p(1, -1) = \frac{4}{81}$  at  $Kuv|_{53}$ .

In (3), i.e. row 2 of matrix  $K_{uv}$ , we follow the curvature distribution for  $u = -2$  fixed. We observe that  $K_p|_{u=-2}$  has small values. So the surface is along  $u = -2$  only minimal curved. In line  $u = 0$  the curvature has a maximal value of 4: this is at the bottom of the surface, where the surface is obviously most curved. All this is in accordance with our intuitive view.

You may follow the coordinate line  $u = -2$  on the u-by-v-region in CALC PLOT3D by tracing the movable point with your mouse to verify the observations above.

**Remark.** In an analogous way, preparing and filling the grid  $H_{uv}$ , one gets the value distribution of the mean curvature  $H$ :

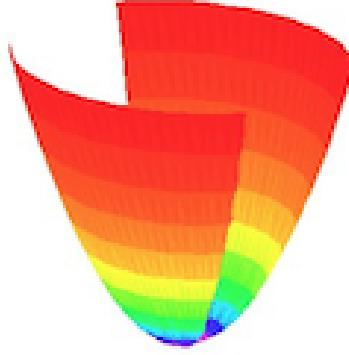
	0	-2	-1	0	1	2
(%o101)	-2	0.35871	0.45722	0.5136	0.45722	0.35871
	-1	0.45722	0.74074	1.0733	0.74074	0.45722
	0	0.5136	1.0733	4.0	1.0733	0.5136
	1	0.45722	0.74074	1.0733	0.74074	0.45722
	2	0.35871	0.45722	0.5136	0.45722	0.35871

d: We construct the tableau of the values  $K_p$  for 5 different user input points  $p$ , where the first column is filled with the headings.

```
"... function table for K with free choosen points";
Ktable : matrix(
  ["u      :" ,1,3.1,-1,0,5],
  ["v      :" ,2, 4, 5,0,5],
  ["Kp(u,v):" ,Kp(1,2), Kp(3.1,4), float(Kp(-1,5)), Kp(0,0), Kp(5,5)]);
```

MAXIMA output:

	<i>... function table for K with free choosen points</i>					
	<i>u</i>	1	3.1	-1	0	5
(Ktable)	<i>v</i>	2	4	5	0	5
	<i>Kp(u,v):</i>	$\frac{4}{441}$	$3.7384 \cdot 10^{-4}$	$3.6281 \cdot 10^{-4}$	4	$\frac{4}{40401}$



*K:* Distribution of the values of Gauss curvature  $K$  using Figure 11: a color code. Equal color means equal  $K$ . Blue color means higher value of  $K$ , red smaller values. – The plot was done with MAPLE VR4 using RECKZIEGEL's `diffgeo` packet.

*In summa: we are now able to calculate the GAUSS  $K$  and mean curvature  $H$  at individual point on a surface, for a sequence (1D vector) of points or a full 2D tableau (matrix) of points to observe the distribution of the  $K$  resp.  $H$  values.*

**Exercise 82.** Solve the previous example using the parametrization (6.9).

For your exploration choose  $a = b = c = 1$ .

a. Verify the

Coefficients of	<i>metric tensor g</i>	<i>2<sup>nd</sup> fundamental tensor h</i>
$E = 1 + \frac{1}{4u}$		$M = \frac{1}{2u\sqrt{1+4u}}$
$F = 0$		$N=0$
$G = u$		$N = \frac{2u}{\sqrt{1+4u}}$

b. Verify:  $K = \frac{4}{(1+4u)^2}$

### 6.3 Exercises

- You may cross-check your results of the following exercises in [18].

**Exercise 83.** (the GAUSS, mean and principal curvature of the *saddle*  $z = xy$ )

A parametrization for this function surface  $S$  is  $X: U \rightarrow \mathbb{R}^3$  with  $(u, v) \mapsto (u, v, u \cdot v)$ .

- Use MAXIMA to calculate the GAUSSian and mean curvature for the saddle.
- Calculate the principal curvatures and the corresponding principal curvature directions.
- Calculate the GAUSSian and mean curvature at the point  $p = (1, 1)$ .

**Exercise 84.** (GAUSS, mean and principal curvature of the hyperbolic paraboloid)

The *hyperbolic paraboloid* is the parametric surface defined by  $X(u, v) = (u, v, u^2 - v^2)$ .

$$\text{Recall: } g = \begin{bmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{bmatrix} = \begin{bmatrix} E & F \\ F & G \end{bmatrix} = \begin{bmatrix} 1+4u^2 & -4uv \\ -4uv & 1+4v^2 \end{bmatrix}$$

$$h = \begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{bmatrix} = \begin{bmatrix} L & M \\ M & N \end{bmatrix} = \begin{bmatrix} 2(4u^2+4v^2+1)^{-1/2} & 0 \\ 0 & -2(4u^2+4v^2+1)^{-1/2} \end{bmatrix}.$$

- Use MAXIMA to calculate the Gaussian and mean curvature functions.
- Calculate GAUSSian and mean curvature at the points  $p_i = (i, 5/i)$ ,  $i \in \{1, 2, 3, 4, 5\}$ .
- Calculate the principal curvatures and the corresponding principal curvature directions.

**Exercise 85.** (GAUSS, mean and principal curvature of the  $\text{?}$  surface)

The  $\text{? surface}$  is the parametric surface defined by  $X(u, v) = (u + v, u - v, uv)$ .

- Determine the 1<sup>st</sup> and 2<sup>nd</sup> fundamental tensors  $g$  and  $h$ .
- Verify: at  $p = (1, 1)$  we have  $K = -1/16$  and  $H = 1/16\sqrt{2}$ .
- Calculate GAUSSian and mean curvature at the points  $p_i = (i, i^2)$ ,  $i \in \{-2, -1, 0, 1, 2\}$ .
- Calculate the principal curvatures and the corresponding principal curvature directions.

**Exercise 86.** (GAUSS, mean and principal curvature of torus)

The *torus* is the parametric surface defined by  $-R = r_o$  in the figure –

$$X(u, v) = ((R + r \cos(v)) \cos(u), (R + r \cos(v)) \sin(u), r \sin(v))$$

with patch region  $(u, v) \in \mathbb{R}^2$ .

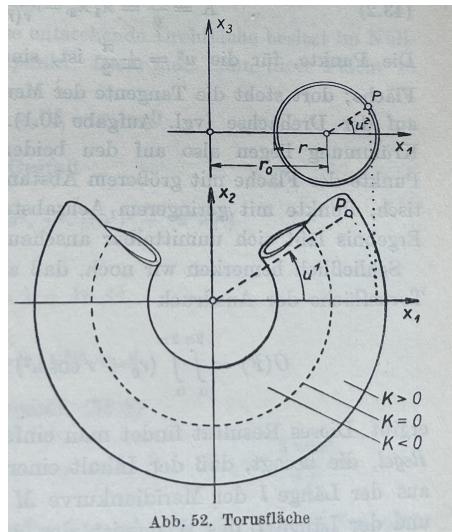


Abb. 52. Torusfläche

KREYSZIG [15, p.165]

Recall: the metric tensor  $g$  has coefficients  $E = (R + r \cos(v))^2, F = 0, G = r^2$ .

the 2<sup>nd</sup> fundamental tensor  $h$  has  $L = -(R + r \cos(v)) \cos(v), M = 0, N = -r$ .

a. Verify:  $K = \frac{\cos(v)}{r(R+r \cos(v))}$ .

b. Calculate  $K(p_i)$  and  $H(p_i)$  at the points  $p_i = (i, i^2), i \in \{-2, -1, 0, 1, 2\}$ .

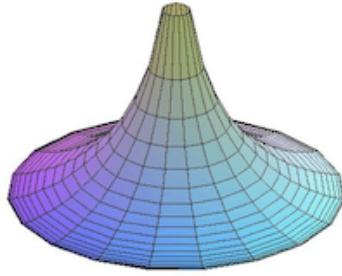
c. Calculate the principal curvatures and the corresponding principal curvature directions.

**Exercise 87.** (GAUSS, mean and principal curvature of BELTRAMI's pseudosphere)

BELTRAMI's pseudosphere is the parametric surface defined by

$$X(u, v) = (\sin u \cos v, \sin u \sin v, \cos u + \ln(\sin u(\cos u + 1)))$$

with patch region  $(u, v) \in \mathbb{R}^+$ .



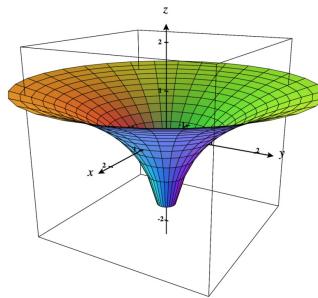
a. Determine the 1<sup>st</sup> and 2<sup>nd</sup> fundamental tensors  $(g_{ij})$  resp.  $(h_{ij})$ .

b. Calculate  $K(p_i)$  and  $H(p_i)$  at the points  $p_i = (i, i^2), i \in \{-2, -1, 0, 1, 2\}$ .

c. Calculate the principal curvatures and the corresponding principal curvature directions.

**Exercise 88.** (GAUSS, mean and principal curvature of a funnel, cf. [2, p.156])

The *funnel* is the parametric surface defined by  $X(u, v) = (v \cos(u), v \sin(u), \ln(v))$ , with patch region  $(u, v) \in [0, 2\pi] \times \mathbb{R}^+$ .



a. Plot the funnel using CALCPLT3D or MAXIMA.

b. Determine the 1<sup>st</sup> and 2<sup>nd</sup> fundamental tensors  $g$  resp.  $h$ .

$$\text{Verify: } g = \begin{bmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{bmatrix} = \begin{bmatrix} v^2 & 0 \\ 0 & 1+v^2 \end{bmatrix}.$$

c. Calculate  $K(p_i)$  and  $H(p_i)$  at the points  $p_i = (i, i^2), i \in \{-2, -1, 0, 1, 2\}$ .

d. Calculate the principal curvatures  $\kappa_i$ .

- You may cross-check your results of the following exercises in [?].
- Calculate the GAUSS curvature  $K$ , the mean curvature  $H$  and principal curvatures  $\kappa_i$  for the following 2D surfaces using MAXIMA ...

**Exercise 89.** the elliptic cylinder  $X(u, v) = (a \cos(u), b \sin(u), v)$  over  $[0, 2\pi] \times \mathbb{R}$ .

Recall:  $g$  has the components  $E = a^2 \sin^2 u + b^2 \cos^2 u, F = 0, G = 1$ .

$h$  has the components  $L = \frac{-ab}{\sqrt{a^2 \sin^2 u + b^2 \cos^2 u}}, M = 0, N = 0$ .

a. Verify:  $K = 0$  and  $H = \frac{-ab}{(a^2 \sin^2 u + b^2 \cos^2 u)^{3/2}}$ .

b. Verify:  $\kappa_1 = \frac{-ab}{(a^2 \sin^2 u + b^2 \cos^2 u)^{3/2}}$  and  $\kappa_2 = 0$ .

**Exercise 90.** the cone  $X(u, v) = (u \cos(v), u \sin(v), u)$  with  $u > 0$ .

Recall:  $g$  has the components  $E = 2, F = 0, G = u^2$ .

$h$  has the components  $L = \frac{u}{\sqrt{2}}, M = 0, N = 0$ .

a. Verify:  $K = 0$  and  $H = \frac{\sqrt{2}}{4u}$ .

b. Verify:  $\kappa_1 = 0$  and  $\kappa_2 = 2H$ .

**Exercise 91.** the monkey saddle  $X(u, v) = (u, v, u^3 - 3uv^2)$ .

Recall:  $g$  has the components  $E = 1 + 9(u^2 - v^2)^2, F = 18uv(v^2 - u^2), G = 1 + 36u^2v^2$ .

Define for the moment:  $a := 1 + 9(u^2 + v^2)^2$ . Then

$h$  has the components  $L = \frac{6u}{\sqrt{a}}, M = \frac{-6v}{\sqrt{a}}, N = \frac{-6u}{\sqrt{a}}$ .

Verify:  $K = \frac{-36(u^2+v^2)}{a^2}$  and  $H = \frac{27u(-u^4+2u^2v^2+3v^4)}{a^{3/2}}$ .

**Exercise 92.** the unit sphere  $X(u, v) = (1 \cos(u) \sin(v), 1 \sin(u) \sin(v), 1 \cos(v))$ .

**Exercise 93.** the catenoid  $X(u, v) = (c \cosh \frac{v}{c} \cos u, c \cosh \frac{v}{c} \sin u, v)$  with  $c = 1$ .

**Exercise 94.** the function graph  $X(u, v) = (u, v, f(u, v))$ .

**Exercise 95.** the surface of revolution  $X(u, v) = (f(u) \cos(v), f(u) \sin(v), g(u))$ .

**Exercise 96.** the PLÜCKER conoid  $X(u, v) = (v \cdot \cos(u), v \cdot \sin(u), 2 \cos(u) \cdot \sin(u))$ .

**Exercise 97.** the SCHERK surface  $X(u, v) = (u, v, \ln(\frac{\cos v}{\cos u}))$ .

**Exercise 98.** the ENNEPER  $X(u, v) = (u - \frac{1}{3}u^3 + uv^2, v - \frac{1}{3}v^3 + vu^2, u^2 - v^2)$ .

**Exercise 99.** the hexenhut  $X(u, v) = (\alpha/\sqrt{u} \cdot \cos v, \alpha/\sqrt{u} \cdot \sin v, u)$  where  $\alpha^2 = \frac{2}{3\sqrt{3}}$ .

**Exercise 100.** the BIANCHI surface

$$X(u, v) = \left( \frac{2 \cdot \sqrt{v^2 + 1} \cdot \sin(u) \cdot \cos(-v + \arctan(v))}{1 + v^2 \sin(u)^2}, \frac{-2 \cdot \sqrt{v^2 + 1} \cdot \sin(u) \cdot \sin(-v + \arctan(v))}{1 + v^2 \sin(u)^2}, \ln(\tan(\frac{1}{2u})) + \frac{2 \cdot \cos(u)}{1 + v^2 \sin(u)^2} \right)$$

## 7 CHRISTOFFEL Symbols and RIEMANN Tensors

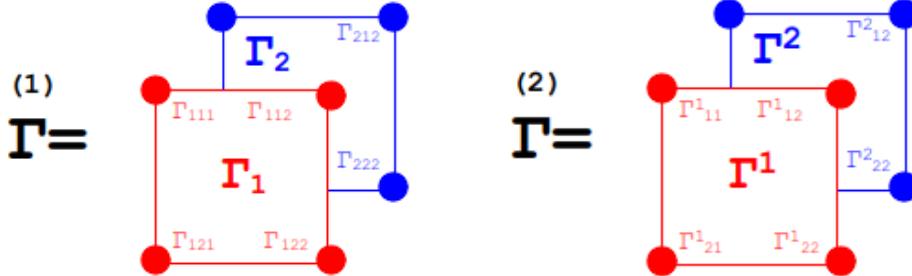


Figure 12: How to imagine the CHRISTOFFEL symbol of first kind  $\Gamma^{(1)}$  and the CHRISTOFFEL symbol of second kind  $\Gamma^{(2)}$  as  $2^3 = 8$  numbers arranged in a tensorial manner in 2 layers.  
This structure is respected by the output of MAXIMA.

The CHRISTOFFEL symbols play a prominent role in Differential Geometry and General Relativity, where we define the RIEMANN curvature tensor  $R^a_{bcd}$ , the RICCI tensor  $R_{bd}$  and the RIEMANN curvature scalar  $R$  with the help of the CHRISTOFFEL symbols of 1<sup>st</sup> and 2<sup>nd</sup> kind. We will need the RIEMANN curvature tensor  $R^a_{bcd}$  to formulate the famous GAUSSIAN *theorema egregium*.

### 7.1 Definition

Let  $X: U \rightarrow \mathbb{R}^3$  be a parametrization of a 2-dimensional parametric surface  $S$ .

For the purpose of this definition we need to heavily index the parameters in a tensorial manner and therefore we use the notation  $(u^1, u^2) \in U$  instead of  $(u, v)$  as before.

Let  $g = (g_{ij})$  be the surface metric tensor of  $X$  with inverse matrix  $g^{-1} = (g^{ij})$ .

a. The CHRISTOFFEL symbol of 1<sup>st</sup> kind, denoted  $\Gamma_{ij,k}: U \rightarrow \mathbb{R}$  is defined by

$$\Gamma_{ij,k} := \frac{\partial^2 X}{\partial u_i \partial u_j} \bullet \frac{\partial X}{\partial u_k} \equiv X_{u^i u^j} \bullet X_{u^k} \quad (7.1)$$

where  $i, j, k \in \{1, 2\}$ <sup>8</sup>.

Using the fundamental metric tensor  $g$  we may write equation (7.1) in equivalent form<sup>9</sup>

$$\Gamma_{ij,k} := \frac{1}{2} \cdot (g_{jk,i} + g_{ik,j} - g_{ij,k}) \quad (7.2)$$

where  $g_{ij,k} \equiv \frac{\partial g_{ij}}{\partial u^k}$  denotes the partial derivative of  $g_{ij}$  with respect to the coordinate  $u^k$ .

<sup>8</sup>cf. BANCHOFF [2, p.220 ff], COXETER [5, p.439], PRESSLEY [24, p.241], SOCHI [32, p.34].

<sup>9</sup>cf. BANCHOFF [2, p.220 ff], COXETER [5, p.439], RECKZIEGEL [25, p.146] or SOCHI [32, p.34].

b. The **CHRISTOFFEL symbol of 2<sup>nd</sup> kind, denoted  $\Gamma^k_{ij}$** :  $U \rightarrow \mathbb{R}$  is defined by using the inverse metric tensor  $g^{ij}$  and memorizing the EINSTEIN summation convention<sup>10</sup> as

$$\Gamma^k_{ij} := \frac{1}{2} \cdot g^{km} \cdot (g_{mi,j} + g_{mj,i} - g_{ij,m}) \quad (7.3)$$

- Decoding the EINSTEIN summation convention and to prepare for the implementation of these formulas in MAXIMA we write equation (7.3) explicit as

$$\Gamma^k_{ij} = \frac{1}{2} \cdot \sum_{m=1}^2 g^{km} \cdot (g_{mi,j} + g_{mj,i} - g_{ij,m}) \quad (7.4)$$

Now we are able to define the RIEMANN curvature tensors of 1<sup>st</sup> and 2<sup>nd</sup> kind. We denote the partial derivative with respect to the coordinate  $u^j$  in short by  $\partial_i$ , because we then may write the formulas for  $R$  in a better memorizable shape, cf. [23, p.91]:

- c. The **RIEMANN curvature tensor of 1<sup>st</sup> kind, denoted  $R_{ijkl}$** :  $U \rightarrow \mathbb{R}$  is defined by

$$R_{ijkl} := \partial_k \Gamma_{jli} - \partial_l \Gamma_{jki} + \Gamma_{ilr} \cdot \Gamma_{jk}^r - \Gamma_{ikr} \cdot \Gamma_{jl}^r \quad (7.5)$$

and the **RIEMANN curvature tensor of 2<sup>nd</sup> kind, denoted  $R^i_{jkl}$**  is defined by

$$R^i_{jkl} := \partial_k \Gamma_{jil}^i - \partial_l \Gamma_{jik}^i + \Gamma_{jl}^r \cdot \Gamma_{rk}^i - \Gamma_{jk}^r \cdot \Gamma_{rl}^i \quad (7.6)$$

### Remark.

- We remark that the latin indices  $i, j, k, l$  used in the previous equations do not imply that we are in 3D ambient space, but are meant here to denote the indexed parameters of the parametric surface, cf. chapter 1.
- The CHRISTOFFEL symbol of 2<sup>nd</sup> kind may also be defined – decoded with the EINSTEIN summation convention – by

$$\Gamma_{ijk} := g_{li} \cdot \Gamma^l_{jk} \quad (7.7)$$

This last definition is an example of lowering the index  $l$  by multiplying with the metric tensor  $g$  ('index juggling').

- We have

LEXICON	<i>Math</i>	<i>MAXIMA</i> (this book)	<i>ctensor</i> (package)
metric tensor $g$ :	$g_{ij}$	$g$	$\ell g$
inverse of metric tensor $g^{-1}$ :	$g^{ij}$	$\text{ginv}$ or $\text{gu}$	$\text{ug}$
CHRISTOFFEL symbol of 1 <sup>st</sup> kind:	$\Gamma_{ijk}$	$\ell\text{Chr}$ or $\ell C$ or $\text{ChrI}$	$\ell cs$
CHRISTOFFEL symbol of 2 <sup>nd</sup> kind:	$\Gamma^k_{ij}$	$\text{uChr}$ or $\text{uC}$ or $\text{ChrII}$	$mcs$
RIEMANN curvature tensor of 1 <sup>st</sup> kind:	$R_{ijkl}$	$\ell R$ or $RI$	$\ell riemann$ or $\ell riem$
RIEMANN curvature tensor of 2 <sup>nd</sup> kind:	$R^i_{jkl}$	$\text{uR}$ or $RII$	$uriemann$ or $uriem$

<sup>10</sup>see e.g. LINDNER [16]

*Recall:* in MAXIMA we encode the tensorial position of the first index as 'upper index' with '**u**' and 'lower index' with '**l**' in the notations of the CHRISTOFFEL symbols  $\Gamma_*^*$  and the RIEMANN curvature tensors  $R_*^*$ . This convention is oriented to V. TODT's **ctensor** package for MAXIMA, which is very useful in this context.

4. The RIEMANN curvature tensor  $R$  of type  $(4, 0)$  resp.  $(3, 1)$  has  $2^4 = 16$  coefficients arranged in 4 2-by-2 matrices, analogue to figure 12.
5. As for equations (7.5) and (7.6) see for example KREYSZIG [15, p.176] , LIPSCHUTZ [20, p.214 ff; 224] , STEEB [35, p.62] or THORPE [37, p.230 ff].
6. Because all objects  $\Gamma_{ijk}$ ,  $\Gamma_{ij}^k$ ,  $R_{ijkl}$ ,  $R_{jkl}^i$  are constructed using only the coefficients of metric tensor  $g_{ij}$ , these objects belongs to the so-called *intrinsic geometry* of surface  $S$ .



Implementing formulas (7.1) ... (7.6) in MAXIMA and calculating some examples should enhance a clear understanding of these mathematical constructs. A CAS like MAXIMA will relieve us of the tedious calculations, which are involved in the terms for the CHRISTOFFEL symbols and the RIEMANN curvature tensors.



## 7.2 Example - $\Gamma$ and $R^i_{jkl}$ for the saddle surface $z = xy$

We now demonstrate two implementations of the CHRISTOFFEL symbols and the RIEMANN curvature tensors in MAXIMA – closely oriented on formulae **a.** resp. **b.** and its term structure. The first implementation use simple user defined functions to reflect the structure of the definition of the concepts and then use MAXIMAs repeat control structure **for** to make the output more comfortable. The second implementation uses clever compact build-in tensor techniques, which were coded by V. TODT in the package **ctensor**, see below.

### 7.2.1 Calculation of $\Gamma$ and $R$ using functions

We show first the implementation and comment it afterwards:

```
/* MAXIMA - DiffGeo14.txt */
"CHRISTOFFEL tensors for the saddle";
define( X(u,v) , [u,v, u*v]);
U : [u,v];

"surface metric tensor";
gij(i,j) := diff(X(u,v),U[i]) . diff(X(u,v),U[j]);      /* (1) */
g : matrix( [gij(1,1),gij(1,2)], [gij(2,1),gij(2,2)] ); /* (2) */
```

```

"CHRISTOFFEL symbol Gamma_ijk of 1st kind using (7.1)";
ChrI(i,j,k) := diff( diff(X(u,v),U[i]), U[j]) . diff(X(u,v), U[k]); /*(3)*/

ChrI(1,2,1);
ChrI(1,2,2);

lChr : matrix( [matrix( [ChrI(1,1,1), ChrI(1,1,2)],      /* (4) */
                         [ChrI(1,2,1), ChrI(1,2,2)] )],
                [matrix( [ChrI(2,1,1), ChrI(2,1,2)],
                         [ChrI(2,2,1), ChrI(2,2,2)] )]);
lChr : radcan(lChr);

for i:1 thru 2 do                                /* (5) */
  for j:1 thru 2 do
    for k:1 thru 2 do
      print([i,j,k],"ChrI:", radcan(ChrI(i,j,k)));

```

- ▷ Paste the code in MAXIMA<sup>on line</sup> or ▷ SAGEMATHCELL

**Comment.** We define the surface via the known parameterization and choose  $U = (u, v)$  as coordinates, where the access  $U[i] \equiv u_i$  in (1) reflects the notion in the text. Equation (1) is the translation of the dot product of the tangential basis vectors  $X_i$  to MAXIMA. In (2) the whole metric is saved in a 2-by-2 matrix named g. Equation (3) use (7.1) to define the CHRISTOFFEL symbol of 1<sup>st</sup> kind, denoted ChrI, *without making use of the metric*. We test this to calculate two values, e.g. ChrI(1,2,1). In (5) we use a for-loop to print all 8 values the CHRISTOFFEL symbol ChrI on the screen. In (4) we save all calculated values of CHRISTOFFEL symbol I in a multi-dimensional matrix ('tensor') named lChr.

wxMAXIMA output:

```

(Clchr)  ⎛ ⎛ 0 0⎞ ⎞
          ⎜ ⎜ v u ⎟ ⎟
          ⎜ ⎜ v u ⎟ ⎟
          ⎜ ⎜ 0 0 ⎟ ⎟
          ⎝ ⎝ ⎠ ⎠
[1,1,1] ChrI: 0
[1,1,2] ChrI: 0
[1,2,1] ChrI: v
[1,2,2] ChrI: u
[2,1,1] ChrI: v
[2,1,2] ChrI: u
[2,2,1] ChrI: 0
[2,2,2] ChrI: 0
(%o74) done

```

- We now implement the CHRISTOFFEL symbol of 2<sup>nd</sup> kind, denoted ChrII. Equation (6) use (7.4): the components are calculated using both the metric  $g$  and the 'contravariant' metric tensor  $gu$  a.k.a. the inverse of the metric tensor in (1). All values are collected in tensor (multimatrix) uChr, see (7).

```

"..." ;
gu : invert(g);      /* the inverse of the metric */
gu : radcan(gu);

"CHRISTOFFEL symbol Gamma^i_jk of 2nd kind using (7.3)";
ChrII(k,i,j) := 1/2* sum( gu[k,m]*( diff(g[m,i],U[j]) +
                                diff(g[m,j],U[i]) - diff(g[i,j],U[m]) ), m,1,2);

ChrII(2,1,2);

uChr : matrix([matrix( [ChrII(1,1,1), ChrII(1,1,2)],                  /* (7) */
                        [ChrII(1,2,1), ChrII(1,2,2)] )],
              [matrix( [ChrII(2,1,1), ChrII(2,1,2)],
                        [ChrII(2,2,1), ChrII(2,2,2)] )]);
uChr : radcan(uChr);

for i:1 thru 2 do                                         /* (8) */
  for j:1 thru 2 do
    for k:1 thru 2 do
      print([i,j,k],"ChrII:", radcan(ChrII(i,j,k)));

```

wxMAXIMA output:

(uChr)	$\begin{pmatrix} 0 & \frac{v}{\sqrt{v^2+u^2+1}} \\ \frac{v}{\sqrt{v^2+u^2+1}} & 0 \end{pmatrix}$ $\begin{pmatrix} 0 & \frac{u}{\sqrt{v^2+u^2+1}} \\ \frac{u}{\sqrt{v^2+u^2+1}} & 0 \end{pmatrix}$
	[1,1,1] ChrII: 0
	[1,1,2] ChrII: $\frac{v}{\sqrt{v^2+u^2+1}}$
	[1,2,1] ChrII: $\frac{v}{\sqrt{v^2+u^2+1}}$
	[1,2,2] ChrII: 0
	[2,1,1] ChrII: 0
	[2,1,2] ChrII: $\frac{u}{\sqrt{v^2+u^2+1}}$
	[2,2,1] ChrII: $\frac{u}{\sqrt{v^2+u^2+1}}$
	[2,2,2] ChrII: 0
(%o82)	done

- As for the RIEMANN curvature tensors of 1<sup>st</sup> and 2<sup>nd</sup> kind  $R_{ijkl}$  and  $R^i_{jkl}$ , denoted `RI(i,j,k,l)` resp. `RII(i,j,k,l)` in the MAXIMA code, the implementation in equations (9) and (11) is straight forward, using the user defined functions `ChrI` and `ChrII` to calculate the CHRISTOFFEL tensors of 1st and 2nd kind. In (10) and (12) we organize a print loop to display the values of the RIEMANN curvature tensors:

```

"RIEMANN curvature tensor Rijkl of 1st kind as function using (7.5)";
RI(b,m,s,q) := diff(ChrI(b,m,q),U[s]) -                               /* (9) */
               diff(ChrI(b,m,s),U[q]) +
               sum(ChrI(b,n,s)*ChrII(n,m,q) ,n,1,2) -
               sum( ChrI(b,n,q)*ChrII(n,m,s) ,n,1,2);

RI : radcan(RI);
radcan(RI(1,2,1,2));                                                 /* (10) */

for b:1 thru 2 do
  for m:1 thru 2 do
    for s:1 thru 2 do
      for q:1 thru 2 do
        print([b,m,s,q] , "RI:", radcan(RI(b,m,s,q)));


"RIEMANN curvature tensor Ri^jkl of 2nd kind as function using (7.6)";
RII(b,m,s,q) :=  diff(ChrII(b,m,q),U[s]) -                           /* (11) */
                  diff(ChrII(b,m,s),U[q]) +
                  sum( ChrII(b,n,s)*ChrII(n,m,q) , n,1,2 ) -
                  sum( ChrII(b,n,q)*ChrII(n,m,s) , n,1,2 );

RII : radcan(RII);
radcan(RII(1,2,1,2));                                                 /* (12) */

for b:1 thru 2 do
  for m:1 thru 2 do
    for s:1 thru 2 do
      for q:1 thru 2 do
        print([b,m,s,q] , "RII:", radcan(RII(b,m,s,q)));

```

wxMAXIMA output:

<pre>(%o86) -<math>\frac{u^2}{v^2+u^2+1}</math> [1,1,1,1] RI: 0 [1,1,1,2] RI: <math>\frac{u v}{v^2+u^2+1}</math> [1,1,2,1] RI: <math>-\frac{u v}{v^2+u^2+1}</math> [1,1,2,2] RI: 0 [1,2,1,1] RI: 0 [1,2,1,2] RI: <math>-\frac{u^2}{v^2+u^2+1}</math> [1,2,2,1] RI: <math>\frac{u^2}{v^2+u^2+1}</math> [1,2,2,2] RI: 0 [2,1,1,1] RI: 0 [2,1,1,2] RI: <math>\frac{v^2}{v^2+u^2+1}</math> [2,1,2,1] RI: <math>-\frac{v^2}{v^2+u^2+1}</math> [2,1,2,2] RI: 0 [2,2,1,1] RI: 0 [2,2,1,2] RI: <math>-\frac{u v}{v^2+u^2+1}</math> [2,2,2,1] RI: <math>\frac{u v}{v^2+u^2+1}</math> [2,2,2,2] RI: 0 (%o87) done</pre>	<pre>(RII) RII (%o91) <math>-\frac{u^2+1}{v^4+(2 u^2+2) v^2+u^4+2 u^2+1}</math> [1,1,1,1] RII: 0 [1,1,1,2] RII: <math>-\frac{u v}{v^4+(2 u^2+2) v^2+u^4+2 u^2+1}</math> [1,1,2,1] RII: <math>\frac{u v}{v^4+(2 u^2+2) v^2+u^4+2 u^2+1}</math> [1,1,2,2] RII: 0 [1,2,1,1] RII: 0 [1,2,1,2] RII: <math>-\frac{u^2+1}{v^4+(2 u^2+2) v^2+u^4+2 u^2+1}</math> [1,2,2,1] RII: <math>\frac{u^2+1}{v^4+(2 u^2+2) v^2+u^4+2 u^2+1}</math> [1,2,2,2] RII: 0 [2,1,1,1] RII: 0 [2,1,1,2] RII: <math>\frac{v^2+1}{v^4+(2 u^2+2) v^2+u^4+2 u^2+1}</math> [2,1,2,1] RII: <math>-\frac{v^2+1}{v^4+(2 u^2+2) v^2+u^4+2 u^2+1}</math> [2,1,2,2] RII: 0 [2,2,1,1] RII: 0 [2,2,1,2] RII: <math>\frac{u v}{v^4+(2 u^2+2) v^2+u^4+2 u^2+1}</math> [2,2,2,1] RII: <math>-\frac{u v}{v^4+(2 u^2+2) v^2+u^4+2 u^2+1}</math> [2,2,2,2] RII: 0 (%o92) done</pre>
---	---

**Remark.** Pay attention: the call of the CHRISTOFFEL symbols or RIEMANN curvature tensors *functions* uses round brackets `(..)`, whereas the direct access to the elements of the corresponding *tensor matrices* uses square brackets `[..]`.

### 7.2.2 Calculation of $\Gamma$ and $R$ with tensor techniques – the `ctensor` package

We now calculate both the CHRISTOFFEL symbols and the RIEMANN curvature tensors using compact tensor techniques with bulid-in functions of the MAXIMA package `ctensor`, designed and implemented by V. TOTH.

```
/* MAXIMA - DiffGeo15.txt */
"CHRISTOFFEL tensors for the saddle";
define( X(u,v) , [u,v, u*v]);
U : [u,v];

gij(i,j) := diff(X(u,v),U[i]) . diff(X(u,v),U[j]);      /* (1) */
g : matrix( [gij(1,1),gij(1,2)], [gij(2,1),gij(2,2)] ); /* (2) */

"CHRISTOFFEL symbols using ctensor:";
load(ctensor)$                                         /* (3) */
ct_coords : [u,v]$;
dim : 2$;
lg : matrix([v^2+1,u*v], [u*v,u^2+1])$           /* see (2) */
cmetric(false)$

"CHRISTOFFEL symbol 1st kind";
christof(lcs); /* lower christoffel symbol */

"CHRISTOFFEL symbol 2nd kind";
christof(mcs); /* mixed (lower and upper indices) christoffel symbol */

"RIEMANN tensor 1st kind";
lriemann(true);

"RIEMANN tensor 2nd kind";
riemann(true);
```

**Comment.** We define the surface metric tensor ( $g_{ij}$ ) via the parameterization  $X$  in (1) and save (2) all metric coefficients in the matrix  $g$ . The next 5 lines *set up a setting*<sup>11</sup> to use the package `ctensor`: we load the package in (3), rename the predefined surface coordinates from  $[x, y]$  to  $[u, v]$ , set the dimension of the surface to 2 and put in the calculated values of metric  $g$  in the *ctensor-variable* `lg`.

The CHRISTOFFEL symbols are now calculated via a call to the `christof()` function with the arguments `lcs` resp. `mcs`.

The RIEMANN curvature tensors are calculated via a call to the `l(eft)riemann()` function resp. the `riemann()` function with its argument set to `true`.

---

<sup>11</sup>Calling the `ctensor` function `csetup()`; after the `load` command in (3), the user is led along the whole setup process in an interactive way.

MAXIMA output:

```
(%o12) CHRISTOFFEL symbol 1st kind
(%t13) lcs1,2,1=v
(%t14) lcs1,2,2=u
(%o14) done
(%o15) CHRISTOFFEL symbol 2nd kind
(%t16) mcs1,2,1= $\frac{v}{v^2+u^2+1}$ 
(%t17) mcs1,2,2= $\frac{u}{v^2+u^2+1}$ 
(%o17) done
(%o18) RIEMANN tensor 1st kind
(%t19) lriem2,2,1,1= $-\frac{1}{v^2+u^2+1}$ 
(%o19) done
(%o20) RIEMANN tensor 2nd kind
(%t21) riem1,2,1,1= $-\frac{u v}{(v^2+u^2+1)^2}$ 
(%t22) riem1,2,1,2= $\frac{1}{v^2+u^2+1} - \frac{u^2}{(v^2+u^2+1)^2}$ 
(%t23) riem2,2,1,1= $-\frac{u^2+1}{v^4+(2 u^2+2) v^2+u^4+2 u^2+1}$ 
(%t24) riem2,2,1,2= $\frac{u v}{v^4+(2 u^2+2) v^2+u^4+2 u^2+1}$ 
(%o24) done
```

- ▷ Paste the code in MAXIMA *on line* – Attention: SAGEMATHCELL does not work here!

**Remark.** 1. Both the CHRISTOFFEL symbols and the RIEMANN curvature tensors are implemented using only the surface metric tensor  $g$ . Hence both objects belongs to the so-called *intrinsic geometry* of the surface  $S = X$ .

2. The CHRISTOFFEL symbol of 2<sup>nd</sup> kind plays a big role in the construction of the so-called LEVI-CIVITA *connection*  $\nabla$  alias the *covariant derivation*  $\nabla_X Y$  of a vector field  $Y$  w.r.t.  $X$ , cf. OLOFF [23, p. 90 ff] or RECKZIEGEL [25, p.147 ff] or THORPE [37, p. 225 ff].

3. The above presented tensorial version of the calculation of CHRISTOFFEL symbol and RIEMANN curvature tensor are most elegant and effective: you only have to set up the necessary requirements. So one may calculate the metric and the derived objects in the same way in higher dimensions, which is essential in General Relativity, see e.g. calculations in the 4D SCHWARZSCHILD space time, cf. OLOFF [23, p. 96 ff] or TOTH [38] ▷ Look for  $i$  Function: cdisplay (ten) or here §9.5, p.94 ff. – Using the user defined for-loop constructs – which are perhaps more simple, insightful and of a didactic surplus – you must therefore manually specify the considered dimension of the space, which was here fixed to 2.

♥ This was a long example. But we wanted to present a gentle rise of abstraction in order to fully understand what is going on and to equip the user which a selection of means to be choose from. The following examples and exercises may therefore be realized according to your mood ♥.

### 7.3 Example - $\Gamma$ and $R^i_{jkl}$ for the Helicoid

Sometimes it is necessary that the user should help MAXIMA to deal with complicated output and to simplify especially terms involving trigonometric functions. This is demonstrated in the following example. Often MAXIMA's `trigsimp` function is very helpful.

**Example 15.** (the calculation of  $\Gamma$  and  $R$  for the helicoid)

The *helicoid* is the parametric surface defined by  $X(u, v) = (v \cos(u), v \sin(u), u)$ .

- Use MAXIMA to calculate the CHRISTOFFEL symbols for the helicoid.
- Use MAXIMA to calculate the RIEMANN curvature tensors for the helicoid.
- Use MAXIMA to calculate the value of  $R^1_{212}$  for the helicoid. Use both the self-written Riemann tensor of the function `RII(1,2,1,2)` and the RIEMANN curvature tensor `riemann` of the `ctensor` package.
- What is the value of `RII(1,2,1,2)|(u,v)=(pi,1)`?

*Solution.* **a:** To determine the CHRISTOFFEL symbols and the RIEMANN curvature tensor we need the surface metric  $g$ . The coefficients of  $g$  should be simplified, before the calculation of the tensors are done - otherwise the results would be very long and uncomfortable and could not easily be crosschecked with the literature.

```
/* MAXIMA - DiffGeo16.txt */
"CHRISTOFFEL tensors for the helicoid";
define( X(u,v) , [v*cos(u), v*sin(u), u]);
U : [u,v]$

"surface metric tensor";
gij(i,j) := diff(X(u,v),U[i]) . diff(X(u,v),U[j])$;
g : matrix( [gij(1,1),gij(1,2)], [gij(2,1),gij(2,2)] );
g : trigsimp(g);                                     /* (1) */

"CHRISTOFFEL symbol of 1st kind";
ChrI(i,j,k) := diff( diff(X(u,v),U[i]), U[j]) . diff(X(u,v), U[k])$;

lChr : matrix([matrix( [ChrI(1,1,1), ChrI(1,1,2)],
                      [ChrI(1,2,1), ChrI(1,2,2)] )],
              [matrix( [ChrI(2,1,1), ChrI(2,1,2)],
                      [ChrI(2,2,1), ChrI(2,2,2)] )]);
lChr : trigsimp(lChr);
```

```

gu : invert(g);      /* the inverse of the metric */

"CHRISTOFFEL symbol of 2nd kind";
ChrII(k,i,j) := 1/2* sum( gu[k,m]*( diff(g[m,i],U[j]) +
diff(g[m,j],U[i]) - diff(g[i,j],U[m]) ), m, 1, 2)$

uChr : matrix([matrix( [ChrII(1,1,1), ChrII(1,1,2)],
[ChrII(1,2,1), ChrII(1,2,2)] )],
[matrix( [ChrII(2,1,1), ChrII(2,1,2)],
[ChrII(2,2,1), ChrII(2,2,2)] )]);

```

- ▷ Click here to RUN this script.

MAXIMA output:

<pre> (%o1)  CHRISTOFFEL tensors for the helicoid (%o2)  X(u,v):=[cos(u)v,sin(u)v,u] (g)   ⎛sin(u)² v²+cos(u)² v²+1   0           0           sin(u)²+cos(u)²⎞           ⎝v²+1   0           0   1⎠ (%o8)  CHRISTOFFEL symbol of 1st kind (lChr) ⎛ 0           -sin(u)² v-cos(u)² v           sin(u)² v+cos(u)² v   0           ⎝sin(u)² v+cos(u)² v   0           0           0⎠ (lChr) ⎛ 0   -v           v   0           v   0           0   0 </pre>	<pre> (gu)   ⎛ 1           ────────── 0           v²+1           0   1 (%o13) CHRISTOFFEL symbol of 2nd kind (uChr) ⎛ 0   v           ──────────           v²+1           v           ⎝-v   0           0   0 </pre>
--	--

- Now we use the MAXIMA tensor package `ctensor` to calculate the CHRISTOFFEL symbols.

```

/* MAXIMA - DiffGeo16.txt */
"CHRISTOFFEL symbols for the helicoid";
"                                     using ctensor:";

load(ctensor)$
ct_coords : [u,v]$ 
dim : 2$ 
lg : matrix([v^2+1,0], [0,1])$ 
cmetric(false)$

christof(false);          /* (1) */

"CHRISTOFFEL symbol 1st kind";
cdisplay(lcs);           /* (2) -- lower christoffel symbol */

"CHRISTOFFEL symbol 2nd kind";
cdisplay(mcs);           /* (3) -- mixed christoffel symbol */

```

▷ Paste the code in MAXIMA<sup>on line</sup>. – Attention: SAGEMATHCELL has no `ctensor`.

wxMAXIMA output:

```

(%o1)  CHRISTOFFEL symbols using ctensor:
(%o7)  done
(%o8)  CHRISTOFFEL symbol 1st kind
        lcs1=
$$\begin{pmatrix} 0 & -v \\ v & 0 \end{pmatrix}$$

        lcs2=
$$\begin{pmatrix} v & 0 \\ 0 & 0 \end{pmatrix}
(%o9)  done
(%o10) CHRISTOFFEL symbol 2nd kind
        mcs1=
$$\begin{pmatrix} 0 & -v \\ v & 0 \end{pmatrix}$$

        mcs2=
$$\begin{pmatrix} v & 0 \\ v^2+1 & 0 \end{pmatrix}
(%o11) done$$$$

```

**Comment.** In (1) we let calculate the CHRISTOFFEL symbols – but do not display them yet (option: `false`). Instead of this we call `cdisplay` in (2) resp. (3) with the arguments `lcs` resp. `mcs` to show both arrays. On the seemingly different index order in (%o10) with regard to `uChr`, see footnote 13, p. 81 below.

*Solution. b, c:* To determine the RIEMANN curvature tensors we need the surface metric  $g$ . The coefficients of  $g$  are simplified in (1) with `trigsimp`, before the calculation of the tensors are done. We use our user-defined functions `RI` and `RII` and alternatively a compact `ctensor` package version.

```

/* MAXIMA - DiffGeo17.txt */
"CHRISTOFFEL tensors for the helicoid";
define( X(u,v) , [v*cos(u), v*sin(u), u]);
U : [u,v]$

"surface metric tensor"
gij(i,j) := diff(X(u,v),U[i]) . diff(X(u,v),U[j])$;
g : matrix( [gij(1,1),gij(1,2)], [gij(2,1),gij(2,2)] );
g : trigsimp(g);                                     /* (1) */

"CHRISTOFFEL symbol of 1st kind"
ChrI(i,j,k) := diff( diff(X(u,v),U[i]), U[j]) . diff(X(u,v), U[k])$

gu : invert(g)$
gu : trigsimp(gu);
ChrII(k,i,j) := 1/2* sum( gu[k,m]*( diff(g[m,i],U[j]) +
diff(g[m,j],U[i]) - diff(g[i,j],U[m]) ), m, 1, 2)$

"RIEMANN curvature tensor Rijkl of 1st kind as function"
RI(b,m,s,q) := diff(ChrI(b,m,q),U[s]) -
diff(ChrI(b,m,s),U[q]) +
sum(ChrI(b,n,s)*ChrII(n,m,q) ,n,1,2) -
sum( ChrI(b,n,q)*ChrII(n,m,s) ,n,1,2)$

for b:1 thru 2 do
  for m:1 thru 2 do
    for s:1 thru 2 do
      for q:1 thru 2 do
        print([b,m,s,q],"RI:", trigsimp(RI(b,m,s,q))));

"RIEMANN curvature tensor Ri^jkl of 2nd kind as function"
RII(b,m,s,q) := diff(ChrII(b,m,q),U[s]) -
diff(ChrII(b,m,s),U[q]) +
sum( ChrII(b,n,s)*ChrII(n,m,q), n,1,2 ) -
sum( ChrII(b,n,q)*ChrII(n,m,s), n,1,2 )$

for b:1 thru 2 do
  for m:1 thru 2 do
    for s:1 thru 2 do

```

```

for q:1 thru 2 do
    print([b,m,s,q],"RII:", trigsimp(RII(b,m,s,q)));
RII(1,2,1,2);          /* solution to c. and d. */
subst([u=%pi, v=1], %);

```

$ \begin{aligned} & [1,1,1,1] \text{ RI: } 0 \\ & [1,1,1,2] \text{ RI: } 0 \\ & [1,1,2,1] \text{ RI: } 0 \\ & [1,1,2,2] \text{ RI: } 0 \\ & [1,2,1,1] \text{ RI: } 0 \\ & [1,2,1,2] \text{ RI: } -\frac{1}{v^2+1} \\ & [1,2,2,1] \text{ RI: } \frac{1}{v^2+1} \\ & [1,2,2,2] \text{ RI: } 0 \\ & [2,1,1,1] \text{ RI: } 0 \\ & [2,1,1,2] \text{ RI: } -\frac{1}{v^2+1} \\ & [2,1,2,1] \text{ RI: } \frac{1}{v^2+1} \\ & [2,1,2,2] \text{ RI: } 0 \\ & [2,2,1,1] \text{ RI: } 0 \\ & [2,2,1,2] \text{ RI: } 0 \\ & [2,2,2,1] \text{ RI: } 0 \\ & [2,2,2,2] \text{ RI: } 0 \\ (\%o15) & \text{done} \end{aligned} $ $ (\%o38) \frac{v^4}{(v^2+1)^2} - \frac{1}{v^2+1} $ $ (\%o39) -\frac{1}{4} $	$ \begin{aligned} & [1,1,1,1] \text{ RII: } 0 \\ & [1,1,1,2] \text{ RII: } 0 \\ & [1,1,2,1] \text{ RII: } 0 \\ & [1,1,2,2] \text{ RII: } 0 \\ & [1,2,1,1] \text{ RII: } 0 \\ & [1,2,1,2] \text{ RII: } -\frac{1}{v^4+2v^2+1} \\ & [1,2,2,1] \text{ RII: } \frac{1}{v^4+2v^2+1} \\ & [1,2,2,2] \text{ RII: } 0 \\ & [2,1,1,1] \text{ RII: } 0 \\ & [2,1,1,2] \text{ RII: } \frac{1}{v^2+1} \\ & [2,1,2,1] \text{ RII: } -\frac{1}{v^2+1} \\ & [2,1,2,2] \text{ RII: } 0 \\ & [2,2,1,1] \text{ RII: } 0 \\ & [2,2,1,2] \text{ RII: } 0 \\ & [2,2,2,1] \text{ RII: } 0 \\ & [2,2,2,2] \text{ RII: } 0 \\ (\%o18) & \text{done} \end{aligned} $
--	---

- Now for the compact `ctensor` package version.

```

/* MAXIMA - DiffGeo17.txt */
"RIEMANN tensors using ctensor:";
load(ctensor)$
ct_coords : [u,v]$ 
dim : 2$ 
lg : matrix([v^2+1,0], [0,1])$ 
cmetric(false)$

ct_coords : [u,v]$ 
dim : 2$ 
lg : matrix([v^2+1,0], [0,1])$ 
cmetric(false)$

```

```

riemann(true);          /* (2) */
cdisplay(riem);        /* (3) */

riem[2,2,1,1];         /* solution to c. and d. */
subst([u=%pi, v=1], %);

```

▷ Paste the code in MAXIMA<sup>on line</sup>. – For this *ctensor* version you can NOT use SAGEMATHCELL!

MAXIMA output:

```

(%t104) riem1,2,1,2=1- $\frac{v^2}{v^2+1}$ 
(%t105) riem2,2,1,1=- $\frac{1}{v^4+2v^2+1}$ 
(%o105) done
riem1,1= $\begin{pmatrix} 0 & 0 \\ 0 & \frac{v^2}{v^2+1}-1 \end{pmatrix}$ 
riem1,2= $\begin{pmatrix} 0 & \frac{1}{v^2+1} \\ 0 & 0 \end{pmatrix}$ 
riem2,1= $\begin{pmatrix} 0 & 0 \\ \frac{1}{v^4+2v^2+1} & 0 \end{pmatrix}$ 
riem2,2= $\begin{pmatrix} -\frac{1}{v^4+2v^2+1} & 0 \\ 0 & 0 \end{pmatrix}$ 
(%o106) done

```

$$\begin{aligned} & (\%o75) -\frac{1}{v^4+2v^2+1} \\ & (\%o76) -\frac{1}{4} \end{aligned}$$

**Comment.** The RIEMANN curvature tensor of 1st kind  $RI = R_{ijkl}$  ist displayed in the first 2 lines as  $riem_{1,2,1,2}$ . The RIEMANN curvature tensor of 2nd kind  $RII = R_{ijkl}$  ist displayed in output line no. (%o105) as  $riem_{1,2}$  etc. c. is solved as special value of  $RII(1,2,1,2) \equiv riem[2,2,1,1]$  in (%o76).

**Remark.** The *ctensor* package version of the RIEMANN curvature tensor is commented in ▷ctensor: 26.2.2 .. riemann. The following index convention is used there:

$$riem[i,j,k,l] \equiv R_{ijk}^l := \Gamma_{ij,k}^l - \Gamma_{ik,j}^l + \Gamma_{mk}^l \cdot \Gamma_{ij}^m - \Gamma_{mj}^l \cdot \Gamma_{ik}^m.$$

Our index convention for the user-defined tensor  $RII$  follows STEEB [35, p.62] et al.:

$$R[b,m,s,q] \equiv R_{mqs}^b := \Gamma_{mq,s}^b - \Gamma_{ms,q}^b + \Gamma_{ns}^b \cdot \Gamma_{mq}^n - \Gamma_{nq}^b \cdot \Gamma_{ms}^n.$$

## 7.4 Exercises

Let's calculate the CHRISTOFFEL symbols and the RIEMANN curvature tensors for our small collection of classical 2D surfaces. Sometimes you should help MAXIMA to simplify trigonometric expressions or to simplify fractions.

- You may cross-check your results of the calculation of the CHRISTOFFEL symbols of the following exercises in [?]. For  $R^I$  and  $R^{II}$  you have to use MAXIMA.
- Calculate the
  - the CHRISTOFFEL symbols  $\Gamma^I$  and  $\Gamma^{II}$
  - the curvature tensor  $R^I$  and
  - the curvature tensor  $R^{II}$  and
  - the value  $\text{RII}(1,2,1,2)$  at  $(u,v) = (\pi, 1)$

for the following surfaces using MAXIMA ...

**Exercise 101.** ... the elliptic cylinder  $X(u,v) = (a \cos(u), b \sin(u), v)$  over  $[0, 2\pi] \times \mathbb{R}$ .

Recall:  $g$  has the components  $E = a^2 \sin^2 u + b^2 \cos^2 u, F = 0, G = 1$ .

Verify:  $\Gamma_{11}^1 = \frac{(a^2 - b^2) \sin(2u)}{2a^2 \sin^2 u + 2b^2 \cos^2 u}$  and all other coefficients are zero.

**Exercise 102.** the cone  $X(u,v) = (u \cos(v), u \sin(v), u)$  with  $u > 0$ .

Recall:  $g$  has the components  $E = 2, F = 0, G = u^2$ .

Verify:  $\Gamma^1 \begin{bmatrix} 0 & 0 \\ 0 & -u/2 \end{bmatrix}$  and  $\Gamma^2 = \begin{bmatrix} 0 & 1/u \\ 1/u & 0 \end{bmatrix}$ .

See [2, p.232].

**Exercise 103.** the elliptic paraboloid  $X(u,v) = (u, v, u^2 + v^2)$ .

**Exercise 104.** the monkey saddle  $X(u,v) = (u, v, u^3 - 3uv^2)$ .

Recall:  $g$  has the components  $E = 1 + 9(u^2 - v^2), F = 18uv(u^2 - v^2), G = 1 + 36u^2v^2$ .

**Exercise 105.** the unit sphere  $X(u,v) = (1 \cos(u) \sin(v), 1 \sin(u) \sin(v), 1 \cos(v))$ .

See ▷ Cox: R.

**Exercise 106.** the catenoid  $X(u,v) = (c \cosh \frac{v}{c} \cos u, c \cosh \frac{v}{c} \sin u, v)$ . Choose  $c = 1$ .

**Exercise 107.** the torus  $X(u,v) = ((b + a \cos(v)) \cos(u), (b + a \cos(v)) \sin(u), a \sin(v))$ .

See [15, p.180, p.383].

**Exercise 108.** the function graph  $X(u,v) = (u, v, f(u,v))$ .

See [15, p.180, p.383].

**Exercise 109.** the surface of revolution  $X(u,v) = (f(u) \cos(v), f(u) \sin(v), g(u))$ .

**Exercise 110.** the PLÜCKER conoid  $X(u,v) = (v \cdot \cos(u), v \cdot \sin(u), 2 \cos(u) \cdot \sin(u))$ .

**Exercise 111.** the SCHERK surface  $X(u,v) = (u, v, \ln(\frac{\cos v}{\cos u}))$ .

Recall:  $g$  has coefficients  $E = \sec^2 u, F = -\tan u \tan v, G = \sec^2 v$ , where  $\sec = \frac{1}{\cos}$ .

Verify:

$$\begin{aligned}\Gamma^1 &= \begin{bmatrix} \tan u & 0 \\ 0 & -\sin u \cos u \sec^2 v \end{bmatrix} \\ \Gamma^2 &= \begin{bmatrix} -\sin v \cos v \sec^2 u & 1 \\ 1 & \tan v \end{bmatrix}\end{aligned}$$

**Exercise 112.** the ENNEPER  $X(u, v) = (u - \frac{1}{3}u^3 + uv^2, v - \frac{1}{3}v^2 + vu^2, u^2 - v^2)$ .  
 Recall: the metric tensor  $g$ :  $E = G = 1 + 2u^2 + 2v^2 + u^4 + 2u^2v^2 + v^4, F = 0$ .

Verify:

$$\Gamma^1 = \begin{bmatrix} \frac{2u}{u^2+v^2+1} & \frac{2v}{u^2+v^2+1} \\ \frac{2v}{u^2+v^2+1} & \frac{-2u}{u^2+v^2+1} \end{bmatrix}$$

**Exercise 113.** the BELTRAMI pseudosphere

$$X(u, v) = (\sin u \cos v, \sin u \sin v, \cos u + \ln(\sin u(\cos u + 1))) .$$

Recall:  $g$  has coefficients  $E = \frac{-\cos^2 u}{\cos^2 u - 1}, F = 0, G = 1 - \cos^2 u$ .

Verify:

$$\Gamma^1 = \begin{bmatrix} \frac{\sin u}{-\cos u + \cos^3 u} & 0 \\ 0 & \frac{\sin u(\cos u - 1)}{\cos u} \end{bmatrix}$$

**Exercise 114.** the hexenhat  $X(u, v) = (\alpha \cdot \frac{\cos v}{\sqrt{u}}, \alpha \cdot \frac{\sin v}{\sqrt{u}}, u)$  where  $\alpha^2 = \frac{2}{3\sqrt{3}}$ .

Recall:  $g$  has coefficients  $E = \frac{1+27u^3}{27u^3}, F = 0, G = \frac{4}{27u}$ .

$$h \text{ has coefficients } (L, M, N) = \left( -\frac{3}{2u^3 \cdot \sqrt{\frac{1+27u^3}{u^4}}}, 0, \frac{2}{u \cdot \sqrt{\frac{1+27u^3}{u^4}}} \right).$$

Verify:

$$\begin{aligned} \Gamma^1 &= \begin{bmatrix} \frac{-3}{2(1+27u^3)} & 0 \\ 0 & \frac{2}{1+27u^3} \end{bmatrix} \\ \Gamma^2 &= \begin{bmatrix} 0 & \frac{-1}{2u} \\ \frac{-1}{2u} & 0 \end{bmatrix} \end{aligned}$$

See: WHEELER [46, p.22 ff]

**Exercise 115.** the 'hexe' surface  $X(u, v) = (\sqrt{u} \cdot \cos v, \sqrt{u} \cdot \sin v, u)$ .

Verify:

a.  $g$  has coefficients  $(E, F, G) = (\frac{1+4u}{4u}, 0, u)$ .

b.  $h$  has coefficients  $(L, M, N) = (\frac{1}{2u \cdot \sqrt{1+4u}}, 0, \frac{2u}{\sqrt{1+4u}})$ .

c.

$$\begin{aligned} \Gamma^1 &= \begin{bmatrix} \frac{-1}{2u(1+4u)} & 0 \\ 0 & \frac{-2u}{1+4u} \end{bmatrix} \\ \Gamma^2 &= \begin{bmatrix} 0 & \frac{1}{2u} \\ \frac{1}{2u} & 0 \end{bmatrix} \end{aligned}$$

**Exercise 116.** the BIANCHI surface

$$X(u, v) = \begin{bmatrix} \frac{2\sqrt{v^2+1} \cdot \sin(u) \cdot \cos(-v+\arctan(v))}{1+v^2 \sin(u)^2} \\ \frac{-2\sqrt{v^2+1} \cdot \sin(u) \cdot \sin(-v+\arctan(v))}{1+v^2 \sin(u)^2} \\ \ln(\tan(\frac{1}{2u})) + \frac{2 \cdot \cos(u)}{1+v^2 \sin(u)^2} \end{bmatrix}$$

**Exercise 117. (The GAUSS Relations for  $\Gamma$ )**

The Christoffel symbols of 2nd kind are given by the following *Gauss relations* as functions of the coefficients  $E, F, G$  of the metric tensor  $g$  and their first order derivations. We have

**Fact.** Let  $X: U \rightarrow \mathbb{R}^3$  be a parametrization of a 2-surface  $S$ .

Let  $g = \begin{bmatrix} X_u \bullet X_u & X_u \bullet X_v \\ X_u \bullet X_v & X_v \bullet X_v \end{bmatrix} \equiv \begin{bmatrix} E & F \\ F & G \end{bmatrix}$  be the metric tensor of  $X$ .

Then we have

$$\Gamma_{11}^1 = \frac{GE_u - 2FF_u + FE_v}{2(EG - F^2)} \quad (7.8)$$

$$\Gamma_{12}^1 = \frac{GE_v - FG_u}{2(EG - F^2)} \quad (7.9)$$

$$\Gamma_{22}^1 = \frac{2GF_v - GG_u - FG_v}{2(EG - F^2)} \quad (7.10)$$

$$\Gamma_{11}^2 = \frac{2EF_u - EE_v - FE_u}{2(EG - F^2)} \quad (7.11)$$

$$\Gamma_{12}^2 = \frac{EG_u - FE_v}{2(EG - F^2)} \quad (7.12)$$

$$\Gamma_{22}^2 = \frac{EG_v - 2FF_v + FG_u}{2(EG - F^2)} \quad (7.13)$$

- a. Program the Gauss relations in MAXIMA for the helicoid.

*Hint:* here is a sketch,  $\Gamma_{11}^1 \parallel_{\text{Math}} = \text{Chr111} \parallel_{\text{Maxima}} \dots$ :

```
/* MAXIMA - DiffGeo18.txt */
"CHRISTOFFEL tensors for the helicoid";
define( X(u,v) , [v*cos(u), v*sin(u), u]);
U : [u,v]$

"surface metric tensor$"
gij(i,j) := diff(X(u,v),U[i]) . diff(X(u,v),U[j])$ 
g : matrix( [gij(1,1),gij(1,2)], [gij(2,1),gij(2,2)] );
g : trigsimp(g);

E : g[1,1];  F: g[1,2];  G: g[2,2];
w : (E*G-F^2);

Chr111 : 1/2* ( G*diff(E,u) - 2*F*diff(F,u) + F*diff(E,v)) / w;
Chr112 : 1/2* ( G*diff(E,v) - 2*F*diff(G,u) ) / w;
"...";
" .. and if you like to have the tensor ChrII";
ChrII1 : matrix([Chr111, Chr112],[Chr121,Chr122]);
ChrII2 : matrix([Chr211, Chr212],[Chr221,Chr222]);

ChrII : matrix([ChrII1], [ChrII2]);
```

**Remark.** A proof is in HECKMANN [13, p.25].

## 8 Intrinsic Curvature

The *intrinsic curvature* of a 2D surface can be calculated using only the coefficients of the metric tensor  $g$  - without recourse to the shape operator who make use of the *extrinsic* concepts of the sff  $h$ .

### 8.1 Definition

Let  $U$  be the parameter region of the metric tensor  $g = (g_{ij}) = g(u, v)$ .

Let  $\partial_u := \frac{\partial}{\partial_u}$  be the partial derivative w.r.t. the variable  $u$  and  $\partial_v := \frac{\partial}{\partial_v}$  ....

Let  $\Gamma^{II} := (\Gamma^1, \Gamma^2)$  be the CHRISTOFFEL symbol of 2<sup>nd</sup> kind associated with  $g$ .

Then we define the *intrinsic curvature*  $K_{intrinsic}$  by

$$K_{intrinsic}(u, v) := \partial_v \Gamma_{11}^2 - \partial_u \Gamma_{21}^2 + \Gamma_{21}^2 \cdot (\Gamma_{11}^1 - \Gamma_{12}^2) + \Gamma_{11}^2 \cdot (\Gamma_{22}^2 - \Gamma_{21}^1) \quad (8.1)$$

**Remark.** 1. The definition (8.1) is in RECKZIEGEL [25, p. 160].

2. In §9 we show, how the GAUSS curvature  $K$  can be calculated using the intrinsic curvature  $K_{intrinsic}$ .

3. Verify:  $K_{intrinsic} = R_{1212}$

### 8.2 Implementation and Examples

We now give two examples and some exercises to calculate the intrinsic curvature  $K_{intrinsic}$ . We start with the so-called hyperbolic half plane, whose results are clear and short and allows a focus on the intermediate steps.

**Example 16.** (The intrinsic curvature of the hyperbolic upper half plane  $H$ )

Let  $H := \{(u, v) \in \mathbb{R}^2 | v > 0\}$  be the *hyperbolic half plane* with 1<sup>st</sup> fundamental form  $ds^2 = \frac{du^2 + dv^2}{v^2}$  resp. its metric tensor  $g(u, v) = \begin{bmatrix} u^2 & 0 \\ 0 & v^2 \end{bmatrix}$ .

Calculate the the intrinsic curvature  $K_{intrinsic}^H(u, v)$  of the hyperbolic half plane  $H$ .

*Solution* We demonstrate two versions.

**1 with user defined functions.** – We use wxMAXIMA. First we compute the CHRISTOFFEL symbols using our user-defined CHRISTOFFEL symbol function. Then we implement the intrinsic curvature by formula (8.1).

```
/* MAXIMA - DiffGeo19a.txt */
"intrinsic curvature for the half plane";
U : [u,v]$

"surface metric tensor"
g : matrix( [1/v^2, 0], [0, 1/v^2] )$
gu : invert(g); /* the inverse of the metric g */
```

```

"CHRISTOFFEL symbol of 2nd kind";
ChrII(k,i,j) := 1/2* sum( gu[k,m]*( diff(g[m,i],U[j]) +
                           diff(g[m,j],U[i]) - diff(g[i,j],U[m]) ), m,1,2)$

Ch111 : ChrII(1,1,1);
Ch121 : ChrII(1,2,1);
Ch211 : ChrII(2,1,1);
Ch221 : ChrII(2,2,1);
Ch222 : ChrII(2,2,2);

define( Kintrinsic(u,v), 1/g[1,1]* (diff(Ch211,v) - diff(Ch221,u) +
                                         Ch221*(Ch111-Ch221) + Ch211*(Ch222-Ch121) ) );
Kintrinsic(u,v);

```

MAXIMA output:

```

(Ch111) 0
(Ch121) -1/v
(Ch211) 1/v
(Ch221) 0
(Ch222) -1/v
(%o184) Kintrinsic(u,v):=-1
(%o185) -1

```

▷ Paste the code in MAXIMA<sup>on line</sup> or ▷ SAGEMATHCELL

**Comment.** The result  $K_{intrinsic}^H(u, v) \equiv -1$  demonstrates, that this curvature is of constant value, because the result is not dependent of both  $u$  and  $v$ . Ergo: The hyperbolic half plane has constant negative intrinsic curvature, cf. BANCHOFF [2, p.227] or RECKZIEGEL [25, p.161].

**2 with ctensor-build-in functions.** – We use wxMAXIMA. First we compute the CHRISTOFFEL symbol of 2<sup>nd</sup> kind, named `mcs[]`<sup>12</sup>, by calling the build-in function `christof`<sup>13</sup> with argument `mcs`. Then we implement the intrinsic curvature by formula (8.1) using the calculated values.

```

/* MAXIMA - DiffGeo19b.txt */
load(ctensor)$
"Intrinsic curvature for the hyperbolic half plane";
ct_coords:[u,v]$
```

<sup>12</sup>`mcs` = mixed Christoffel symbol

<sup>13</sup>Attention: The index convention of `christof` is not the same as in our user defined functions `ChrI/II`. I thank VIKTOR T. TOTH for the hint, that the index convention in `ctensor` is  $\Gamma_{ij}^k = mcs[i, j, k]$ . Therefore the index lists in 2. differs from that in 1.

```

dim:2$
lg : matrix([1/v^2, 0], [0, 1/v^2])$
cmetric(false)$
christof(mcs);

Ch111 : mcs[1,1,1];
Ch121 : mcs[1,2,1];
Ch211 : mcs[1,1,2];
Ch221 : mcs[2,1,2];
Ch222 : mcs[2,2,2];

define( Kintrinsic(u,v), 1/lg[1,1]* (diff(Ch211,v) - diff(Ch221,u) +
                                         Ch221*(Ch111-Ch221) + Ch211*(Ch222-Ch121) ));
Kintrinsic(u,v);

```

MAXIMA output:

(%o2)	<i>Intrinsic curvature for the hyperbolic half plane</i>
(%t7)	$mcs_{1,1,2} = \frac{1}{v}$
(%t8)	$mcs_{1,2,1} = -\frac{1}{v}$
(%t9)	$mcs_{2,2,2} = -\frac{1}{v}$
(%o9)	<i>done</i>
(%o15)	$\text{Kintrinsic}(u,v) := -1$
(%o16)	$-1$

▷ Paste the code in MAXIMA<sup>on line</sup> SAGEMATHCELL does not work!

**Example 17.** (The intrinsic curvature of the helicoid)

The metric tensor  $g$  of the helicoid is known to have the coefficients  $E = 1, F = 0, G = 1 + v^2$ . Calculate the the intrinsic curvature  $K_{intrinsic}^{helicoid}(u, v)$  of the helicoid.

*Solution.*

**1 with user-defined functions.** – We use wxMAXIMA. First we compute the CHRISTOFFEL symbols using our function `ChrII`. Then we calculate the intrinsic curvature by formula (8.1) coded in function `Kintrinsic`, which has the 5 necessary CHRISTOFFEL symbol coefficients put inside the `block` structure as local variables.

```

/* MAXIMA - DiffGeo20a.txt */
"intrinsic curvature for the HELICOID";
U:[u,v]$;
"surface metric tensor"$;
E : 1$;
F : 0$;
G : v^2+1$;

```

```

g : matrix( [E,F], [F,G] )$
gu : invert(g); /* the inverse of the metric */

"CHRISTOFFEL symbol of 2nd kind"
ChrII(k,i,j) := 1/2* sum( gu[k,m]*( diff(g[m,i],U[j]) +
                           diff(g[m,j],U[i]) - diff(g[i,j],U[m]) ), m,1,2)$

define( Kintrinsic(u,v),
        block([Ch111: ChrII(1,1,1), Ch121: ChrII(1,2,1), Ch211: ChrII(2,1,1),
               Ch221: ChrII(2,2,1), Ch222: ChrII(2,2,2)],
              1/g[1,1]* (diff(Ch211,v) - diff(Ch221,u) +
                          Ch221*(Ch111-Ch221) + Ch211*(Ch222-Ch121))));

Kintrinsic(u,v);

```

MAXIMA output:

```

(%o55) Kintrinsic curvature for the HELICOID
          ⎛ 1   0 ⎞
(%o56) ⎝ 0   1 ⎠
(%o65) Kintrinsic(u,v):=0
(%o66) 0

```

▷ Paste the code in MAXIMA<sup>on line</sup> or ▷ SAGEMATHCELL

**2 with ctensor-build-in functions.** – We use wxMAXIMA. First we compute the CHRISTOFFEL symbol of 2<sup>nd</sup> kind `mcs[]` by calling the build-in function `christof` with argument `mcs`. Then we implement the intrinsic curvature by formula (8.1) using the calculated values.

```

/* MAXIMA - DiffGeo20b.txt */
"Intrinsic curvature for the helicoid";
load(ctensor)$
ct_coords:[u,v]$ 
dim:2$

"surface metric tensor"
[E,F,G] : [1,0,v^2+1]$
lg : matrix( [E,F], [F,G] )$
cmetric(false)$

christof(mcs);
define( Kintrinsic(u,v),
        block ([Ch111: mcs[1,1,1], Ch121: mcs[1,2,1], Ch211: mcs[1,1,2],
                Ch221: mcs[2,1,2], Ch222: mcs[2,2,2]],
               1/lg[1,1]* (diff(Ch211,v) - diff(Ch221,u) +
                           Ch221*(Ch111-Ch221) + Ch211*(Ch222-Ch121))));
```

```
Kintrinsic(u,v);
```

MAXIMA output:

```
(%o1) Intrinsic curvature for the helicoid
      v
(%t11) mcs2,2,2=-----v2+1
(%o11) done
(%o12) Kintrinsic(u,v):=0
(%o13) 0
```

▷ Paste the code in MAXIMA *on line* – SAGEMATHCELL does not work!

**Comment.** The result  $K_{intrinsic}^{helicoid}(u, v) \equiv 0$  demonstrates, that the helicoid has constant intrinsic curvature 0 at every point.

### 8.3 Exercises

- You may cross-check your results of the following exercises in [18].
- Task: Calculate the intrinsic curvature  $K_{intrinsic}(u, v)$  for the following surfaces using MAXIMA ...

**Exercise 118.** the cylinder  $X(u, v) = (a \cos(v), a \sin(v), u)$ . Choose  $a = 1$ .

Verify:  $K_{intrinsic}^{cylinder}(u, v) = 0$

**Exercise 119.** the cone  $X(u, v) = (u \cos(v), u \sin(v), u)$ .

Verify:  $K_{intrinsic}^{cone}(u, v) = 0$

**Exercise 120.** the elliptic paraboloid  $X(u, v) = (u, v, u^2 + v^2)$ .

Verify:  $K_{intrinsic}^{ell.parab.}(u, v) = \frac{4}{(4u^2+4v^2+1)^2}$

**Exercise 121.** the hyperbolic paraboloid  $X(u, v) = (u, v, u^2 - v^2)$ .

Verify:  $K_{intrinsic}^{hyp.parab.}(u, v) = \frac{-4}{(4u^2+4v^2+1)^2}$

**Exercise 122.** the monkey saddle  $X(u, v) = (u, v, u^3 - 3uv^2)$ .

Verify:  $K_{intrinsic}^{monkey}(u, v) = -36 \frac{u^2+v^2}{(9u^4+18u^2v^2+9v^4+1)^2}$

**Exercise 123.** the unit sphere  $X(u, v) = (\cos(u) \sin(v), \sin(u) \sin(v), \cos(v))$ .

Verify:  $K_{intrinsic}^{sphere1}(u, v) = 1$

**Exercise 124.** the catenoid  $X(u, v) = (c \cosh \frac{v}{c} \cos u, c \cosh \frac{v}{c} \sin u, v)$ . Choose  $c = 1$ .

Verify:  $K_{intrinsic}^{catenoid}(u, v) = 0$

**Exercise 125.** the torus  $X(u, v) = ((b + a \cos(v)) \cos(u), (b + a \cos(v)) \sin(u), a \sin(v))$ .

Verify:  $K_{intrinsic}^{torus}(u, v) = \frac{\cos v}{2+\cos v}$

**Exercise 126.** the function graph  $X(u, v) = (u, v, f(u, v))$ .

Verify:  $K_{intrinsic}^{graph}(u, v) = \frac{f_{uu}f_{vv}-f_{uv}^2}{(f_u^2+f_v^2+1)^2}$

**Exercise 127.** the surface of revolution  $X(u, v) = (f(u) \cos(v), f(u) \sin(v), g(u))$ .

$$\text{Verify: } K_{intrinsic}^{revolution}(u, v) = \frac{g_u(-f_{uu}g_u + g_{uu}f_u)}{f(u)(f_u^2 + g_u^2)^2}$$

**Exercise 128.** the PLÜCKER conoid  $X(u, v) = (v \cdot \cos(u), v \cdot \sin(u), 2 \cos(u) \cdot \sin(u))$ .

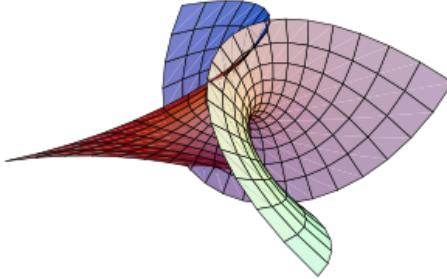
$$\text{Verify: } K_{intrinsic}^{Pluecker}(u, v) = \frac{-4(4 \cos^4 u - 4 \cos^2 u + 1)}{(\cos^4 v + 2 \cos^2 u \cos^2 v - \dots)}$$

**Exercise 129.** the SCHERK surface  $X(u, v) = (u, v, \ln(\frac{\cos v}{\cos u}))$ .

$$\text{Verify: } K_{intrinsic}^{Scherk}(u, v) = \frac{-\cos^2 u \cos^2 v}{(\cos^4 v + 2 \cos^2 u \cos^2 v - \dots)}$$

**Exercise 130.** the ENNEPER  $X(u, v) = (u - \frac{1}{3}u^3 + uv^2, v - \frac{1}{3}v^2 + vu^2, u^2 - v^2)$ .

$$\text{Verify: } K_{intrinsic}^{Enneper}(u, v) = \frac{-4}{(1+2u^2+2v^2+u^4+2v^2u^2+v^4)(u^2+v^2+1)^2}$$



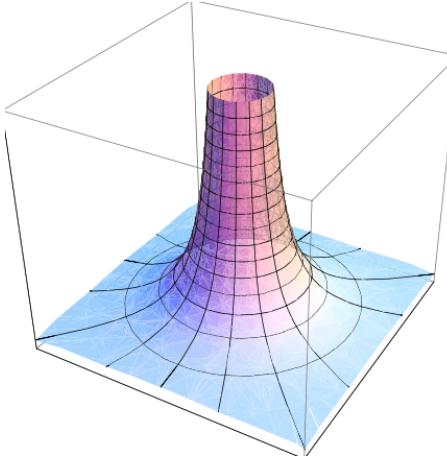
**Exercise 131.** the BELTRAMI pseudosphere

$$X(u, v) = (\sin u \cos v, \sin u \sin v, \cos u + \ln(\sin u(\cos u + 1)))$$

$$\text{Verify: } K_{intrinsic}^{Beltrami}(u, v) = -1$$

**Exercise 132.** the hexenhat  $X(u, v) = (\alpha \cdot \frac{\cos v}{\sqrt{u}}, \alpha \cdot \frac{\sin v}{\sqrt{u}}, u)$  where  $\alpha^2 = \frac{2}{3\sqrt{3}}$ .

$$\text{Verify: } K_{intrinsic}^{hexenhat}(u, v) = \frac{-2187u^4}{(1+27u^3)^2}$$

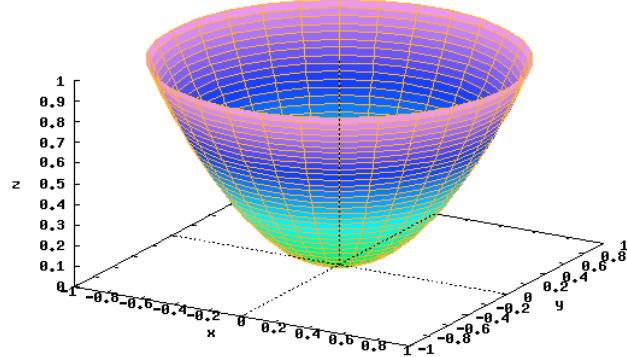


**Exercise 133.** the 'hexe' surface  $X(u, v) = (\sqrt{u} \cdot \cos v, \sqrt{u} \cdot \sin v, u)$ .

Verify:  $K_{intrinsic}^{hexe}(u, v) = \frac{4}{(1+4u)^2}$

- Plot of hexe surface with wxMAXIMA:

```
wxplot3d([sqrt(u)*cos(v),sqrt(u)*sin(v),u],[v,0,2*pi],[u,0,1])$
```



**Exercise 134.** the BIANCHI surface

$$X(u, v) = \begin{bmatrix} \frac{2\sqrt{v^2+1}\sin(u)\cos(-v+\arctan(v))}{1+v^2\sin(u)^2} \\ \frac{-2\sqrt{v^2+1}\sin(u)\sin(-v+\arctan(v))}{1+v^2\sin(u)^2} \\ \ln(\tan(\frac{1}{2u})) + \frac{2\cos(u)}{1+v^2\sin(u)^2} \end{bmatrix}$$

Verify:  $K_{intrinsic}^{Bianchi}(u, v) = -1$

## 9 RICCCI Tensor and GAUSS Theorema egregium

The *intrinsic curvature* of a 2D surface can be calculated using only the coefficients of the metric tensor  $g$  - without recourse to the shape operator who make use of the *extrinsic* concepts of the sff  $h$ .

### 9.1 Definition - RICCI curvature

Let  $X: U \rightarrow \mathbb{R}^2$  be the parameterization of a surface  $S$ .

Let  $g = (g_{ij})$  be the surface metric tensor of  $X$  with inverse matrix  $g^{-1} = (g^{ij})$ .

Let  $R^i_{jkl}$  be the RIEMANN curvature tensor of 2<sup>nd</sup> kind.

Then we define ...

- a. ... the **RICCI tensor**  $\text{Ric}$  via a contraction w.r.t. the middle index of the RIEMANN curvature tensor by

$$\text{Ric}_{ik} := \sum_{j=1}^2 R^j_{ijk} \quad (9.1)$$

where the **EINSTEIN summation convention** is decoded to prepare for the implementation of the formula in MAXIMA.

- b. ... the **mixed RICCI tensor**  $\text{Ric}$  via index juggling of the RICCI tensor by

$$\text{Ric}^j_i := \sum_{k=1}^2 g^{jk} \text{Ric}_{ik} \quad (9.2)$$

- c. ... the **RICCI curvature (scalar)**  $R$  via index contraction of the RICCI tensor by

$$R := \sum_{i=1}^2 \sum_{k=1}^2 g^{ik} \text{Ric}_{ik} \quad (9.3)$$

alias  $\text{R} := \text{Ric}_i^i$ .

**Remark.** The RICCI tensor  $\text{Ric}$  is often also called the *scalar curvature* of the surface  $S$  resp. of the metric  $g$ .

## 9.2 GAUSS's Theorema Egregium

**Theorem. (Gauss 1827)**

LET  $X: U \rightarrow \mathbb{R}^2$  be the parameterization of a surface  $S$ .

Let  $g = (g_{ij})$  be the surface metric tensor of  $X$  with coefficients  $E, F, G$ .

Let  $K$  be the GAUSS curvature and  $K_{intrinsic}$  be the intrinsic curvature of  $X$ .

Let  $R$  be the RICCI curvature scalar.

THEN we have:

**A.** *The GAUSS curvature is half of the RICCI curvature scalar, i.e.*

$$R = 2K \quad (9.4)$$

**B.** *The intrinsic curvature is  $g_{11}$ -times of the GAUSS curvature, i.e.*

$$K_{intrinsic} = K \cdot g_{11} \quad (9.5)$$

**C.** *The GAUSS curvature is explicit calculated via BRIOSCHI's formula from 1852, i.e.*

$$K = \frac{\det \begin{pmatrix} -\frac{1}{2} \frac{\partial^2 E}{\partial v^2} + \frac{\partial^2 F}{\partial u \partial v} - \frac{1}{2} \frac{\partial^2 G}{\partial u^2} & \frac{1}{2} \frac{\partial E}{\partial u} & \frac{\partial F}{\partial u} - \frac{1}{2} \frac{\partial E}{\partial v} \\ \frac{\partial F}{\partial v} - \frac{1}{2} \frac{\partial G}{\partial u} & E & F \\ \frac{1}{2} \frac{\partial G}{\partial v} & F & G \end{pmatrix} - \det \begin{pmatrix} 0 & \frac{1}{2} \frac{\partial E}{\partial v} & \frac{1}{2} \frac{\partial G}{\partial u} \\ \frac{1}{2} \frac{\partial E}{\partial v} & E & F \\ \frac{1}{2} \frac{\partial G}{\partial u} & F & G \end{pmatrix}}{(EG - F^2)^2} \quad (9.6)$$

**Comment.** The main statement of the **Theorema Egregium** from 1827 is: *the GAUSS curvature  $K$  of a surface parameterization is a measure of its intrinsic geometry*, i.e.  $K$  can be calculated of the coefficients  $E, F, G$  of the metric tensor  $g$  alone! This is explicitly stated in the BRIOSCHI formula (9.6).

While the WEINGARTEN map alias the shape operator describes the shape of a surface as part of the ambient space, the numbers  $E, F, G$  and therefore the measurement of length or area on the surface do not take any borrowing from the ambient space.

The RICCI tensor, the RICCI curvature and the RICCI curvature scalar are therefore concepts of the intrinsic geometry of the surface. In contrast the WEINGARTEN map alias the shape operator are concepts of the extrinsic geometry of the surface – by using the normal vector field of  $S$ .

It is the merit of C. F. GAUSS to have shown that the GAUSS curvature can be traced back to the 3 real-valued functions  $E, F, G$  and their derivatives.

**Remark.** 1. A proof of A. is in OLOFF [23, p.110].

2. A proof of B. is in RECKZIEGEL [25, p.160].

3. A proof of C. is in BANCHOFF [2, p.230], HECKMANN [13, p.30] or PRESSLY [24, p.233].

### 9.3 Implementation and Examples

**Example 18.** (Calculation of the RICCI curvature scalar  $R$  of the helicoid)

We do the calculation as before with/without tensor techniques in MAXIMA. The metric tensor  $g$  of the helicoid is known to have the coefficients  $E = 1, F = 0, G = 1 + v^2$ . Calculate the RICCI tensor and RICCI curvature scalar  $R$  of the helicoid.

*Solution*

1. We use MAXIMA. First we define and compute the RICCI curvature tensor  $\text{Ric}$  with our user defined functions and save all values in the container  $\text{RIC}$ . We use formula (9.3) to 'double contract' the  $\text{Ric}$  tensor using  $\text{sum}()$  to get the RICCI curvature scalar  $R$ .

```
/* MAXIMA - DiffGeo20.txt */
"RICCI tensor and scalar curvature for the HELICOID";
U : [u,v]$                      /* surface coordinates */

"surface metric tensor$"
g  : matrix( [1, 0], [0, v^2+1] )$
gu : invert(g)$                  /* the inverse of the metric */

"CHRISTOFFEL symbol 2nd kind$"
ChrII(k,i,j) := 1/2* sum( gu[k,m]*( diff(g[m,i],U[j]) +
                                diff(g[m,j],U[i]) - diff(g[i,j],U[m]) ), m,1,2)$

"RIEMANN curvature tensor Ri^jkl of 2nd kind$"
RII(b,m,s,q) :=  diff(ChrII(b,m,q),U[s]) -
                  diff(ChrII(b,m,s),U[q]) +
                  sum( ChrII(b,n,s)*ChrII(n,m,q), n,1,2) -
                  sum( ChrII(b,n,q)*ChrII(n,m,s), n,1,2)$

"Ricci tensor";
Ric(m,q) := sum( RII(s,m,s,q), s,1,2);

"display Ricci tensor coefficients:";
for m:1 thru 2 do
    for q:1 thru 2 do
        print([m,q],"Ric:", radcan(Ric(m,q)))$

"save Ricci tensor coefficients in matrix RIC:";
RIC : matrix( [Ric(1,1),Ric(1,2)], [Ric(2,1),Ric(2,2)] )$  
RIC;

"Ricci scalar curvature R";
R : sum( sum( gu[i,j]*Ric(i,j) ,j,1,2) ,i,1,2);
```

MAXIMA output:

```
(%o93) Ric(m, q):=  $\sum_{s=1}^r \text{RII}(s, m, s, q)$ 
      [1, 1] Ric: 0
      [1, 2] Ric: 0
      [2, 1] Ric: 0
      [2, 2] Ric: 0
(%o96) 
       $\begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$ 
(%o97) Ricci scalar curvature R
(R) 0
```

▷ Paste the code in MAXIMA<sup>on line</sup> or ▷ SAGEMATHCELL

**Comment.** The result  $R^{helicoid} = 0$  demonstrates, that the helicoid has a vanishing RICCI curvature scalar at every point.

**2.** This time we first compute the RICCI tensor `ric` using compact tensor techniques from `ctensor`. This is not really necessary, because the RICCI curvature scalar  $R$  can be directly computed via a call to the `ctensor` package function `scurvature()`. But we want to demonstrate the call to `ric[]` and to visit the associated matrix ('tensor'). The EINSTEIN summation convention is hidden behind the build-in implementation of `ric` resp. `scurvature`.<sup>14</sup> Then we invoke `scurvature` alias `R` to calculate the RICCI curvature scalar. Therefore this solution is very compact – but hides the definitions of the working functions, see 1.

```
/* MAXIMA - DiffGeo21.txt */
"RICCI curvature for the helicoid";
load(ctensor)$
ct_coords:[u,v]$ 
dim:2$ 
lg : matrix( [1, 0], [0, v^2+1] )$$
cmetric(false)$

"RICCI tensor:";
ricci(true);      /* compute RICCI tensor */
cdisplay(ric);    /* show RICCI tensor */

"RICCI scalar curvature:";
scurvature();     /* = R */
```

▷ Paste the code in MAXIMA<sup>on line</sup> – !SAGEMATHCELL does not work with `ctensor`!

---

<sup>14</sup>In 1. this was explicitly done using summation  $\sum := \text{sum}(\dots)$  || MAXIMA.

MAXIMA output:

```

(%o1)  RICCI curvature for the helicoid
(%o7)  RICCI tensor
      THIS SPACETIME IS EMPTY AND/OR FLAT
(%o8)  done
      ric = 
$$\begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$$

(%o9)  done
(%o10) RICCI scalar curvature
(%o11)  0

```

**Exercise 135.** (BRIOSCHI's formula)

Program BRIOSCHI's formula (9.6) in MAXIMA and compute the GAUSS curvature  $K$  for the helicoid with it.

*Solution:* sketch. Fill in the . . .

```

/* MAXIMA - DiffGeo22.txt */
"Brioschi's formula for the helicoid";
define( X(u,v), [u*cos(v), u*sin(v), c* v] )$
U : [u,v]$           /* surface coordinates */
"metric tensor of helicoid:";
g  : matrix( [1, 0], [0, c^2+u^2] )$
gu : invert(g)$
E:g[1,1]; F:g[1,2]; G:g[1,1];

"define the 2 matrices M1 and M2 as container";
M1 : zeromatrix(3,3)$
M1[1,1] : -1/2*diff(E,v,2) + diff(diff(F,u),v) -1/2*diff(G,u,2);
"...";
M1;

M2 : zeromatrix(3,3)$
M1[1,2] : 1/2*diff(E,v);
"...";
M2;

"Calculate GAUSS curvature K:";
K(u,v) := (determinant(M1) - determinant(M2))/(E*G -F^2);
K(u,v);

```

▷ Paste the code in MAXIMA<sup>on line</sup> or ▷ SAGEMATHCELL

**Exercise 136.** Compute the GAUSS curvature  $K$  for the 'saddle' surface  $z = uv$  with BRIOSCHI's formula.

## 9.4 The sine-Gordon Equation

$$\frac{\partial^2 u}{\partial x \partial y} = \sin(u)$$

This is the so-called *sine-GORDON equation*, which was considered in the study of surfaces of constant Gaussian curvature  $K = -1$ , called *pseudospherical surfaces*. We show how this equation can be derived using MAXIMA. In this example we demonstrate how to get this equation from the metric tensor  $g$  resp. the 1<sup>st</sup> fundamental form  $I$  of a surface.

*Task.* Let  $U \subset \mathbb{R}^2$  be the region of the metric tensor  $g$ , which is given by its 1<sup>st</sup> fundamental form  $I = dx \wedge dx + \cos(u(x, y)) \cdot dx \wedge dy + \cos(u(x, y)) \cdot dy \wedge dx + dy \wedge dy$  alias its metric coefficients  $g_{11} = g_{22} = 1$ ,  $g_{12} = g_{21} = \cos(u(x, y))$ .

Calculate the RICCI curvature scalar  $R$  of  $g$ , i.e. verify

$$R = -\frac{2}{\sin(u)} \frac{\partial^2 u}{\partial x \partial y}$$

*Solution.*

1<sup>st</sup> We compute the RICCI scalar curvature  $R$  using our build-in functions. We simplify the contravariant metric tensor  $\text{Math: } g^{-1}$   $\text{Maxima: } \text{gu}$  and the RICCI curvature scalar with `trigsimp()`.

```
/* MAXIMA - DiffGeo23a.txt */
"RICCI tensor and s-curvature for the sine-GORDAN equation";
U : [u,v]$                      /* surface coordinates */
depends(f,[u,v])$

"surface metric tensor"
g : matrix( [1, cos(f)], [cos(f), 1] )$ 
gu : invert(g);                  /* the inverse of the metric */
gu : trigsimp(gu);

"CHRISTOFFEL symbol 2nd kind"
ChrII(k,i,j) := 1/2* sum( gu[k,m]*( diff(g[m,i],U[j]) +
                           diff(g[m,j],U[i]) - diff(g[i,j],U[m]) ), m,1,2)$

"RIEMANN curvature tensor Ri^jkl of 2nd kind"
RII(b,m,s,q) := diff(ChrII(b,m,q),U[s]) -
                 diff(ChrII(b,m,s),U[q]) +
                 sum( ChrII(b,n,s)*ChrII(n,m,q), n,1,2) -
                 sum( ChrII(b,n,q)*ChrII(n,m,s), n,1,2)$

"Ricci tensor";
Ric(m,q) := sum( RII(s,m,s,q), s,1,2);

"Ricci scalar curvature R";
```

```
R : sum( sum( gu[i,j]*Ric(i,j) ,j,1,2) ,i,1,2);
trigsimp(R);
```

wxMAXIMA output:

$$\begin{aligned} \text{(%)43)} & \quad \text{Ricci scalar curvature } R \\ \text{(R)} & = -\frac{2 \cos(f) \left( \frac{\cos(f)^2 (f_u) (f_v)}{\sin(f)^2} - \frac{(f_u) (f_v)}{\sin(f)^2} + (f_u) (f_v) - \frac{\cos(f) (f_{uv})}{\sin(f)} \right)}{\sin(f)^2} - \frac{2 (f_{uv})}{\sin(f)^3} \\ \text{(%)45)} & = -\frac{2 (f_{uv})}{\sin(f)} \end{aligned}$$

▷ Paste the code in MAXIMA<sup>on line</sup> or ▷ SAGEMATHCELL

2<sup>nd</sup> We now compute the RICCI scalar curvature  $R$  using tensor techniques of the ctensor package. I owe this solution to a correspondence with VIKTOR T. TOTH.

```
/* MAXIMA - DiffGeo23a.txt */
"RICCI tensor and s-curvature for the sine-GORDAN equation";
derivabbrev:true$ ratfac:true$
load(ctensor)$
ct_coords:[u,v]$ 
depends(f,[u,v])$ 
dim:2$ 
lg : matrix( [1, cos(f)], [cos(f), 1] )$ 
cmetric(false)$ 
ric:zeromatrix(2,2)$ 
ricci(false)$ 
ric:trigsimp(expand(ric));
trigsimp(expand(scurvature()));
```

MAXIMA output:

$$\begin{aligned} \text{(%)01)} & \quad \text{RICCI tensor and s-curvature for the sine-GORDAN equation} \\ \text{(ric)} & = \begin{pmatrix} -\frac{f_{uv}}{\sin(f)} & -\frac{\cos(f) (f_{uv})}{\sin(f)} \\ -\frac{\cos(f) (f_{uv})}{\sin(f)} & -\frac{f_{uv}}{\sin(f)} \end{pmatrix} \\ \text{(%)03)} & = -\frac{2 (f_{uv})}{\sin(f)} \end{aligned}$$

▷ Paste the code in MAXIMA<sup>on line</sup> – SAGEMATHCELL does not support ctensor!

**Remark.** 1. Our presentation of the sine-Gordon equation followed STEEB in [36, p. 64–67 and p. 363–365], where an implementation with for-loops is given in REDUCE and C++.  
2. For more info see e.g. ▷ Sine-Gordon equation.

## 9.5 The RICCI scalar of the SCHWARZSCHILD Spacetime Metric

Über das Gravitationsfeld eines Massenpunktes  
nach der EINSTEINSchen Theorie.

Von K. SCHWARZSCHILD.

(Vorgelegt am 13. Januar 1916 [s. oben S. 42].)

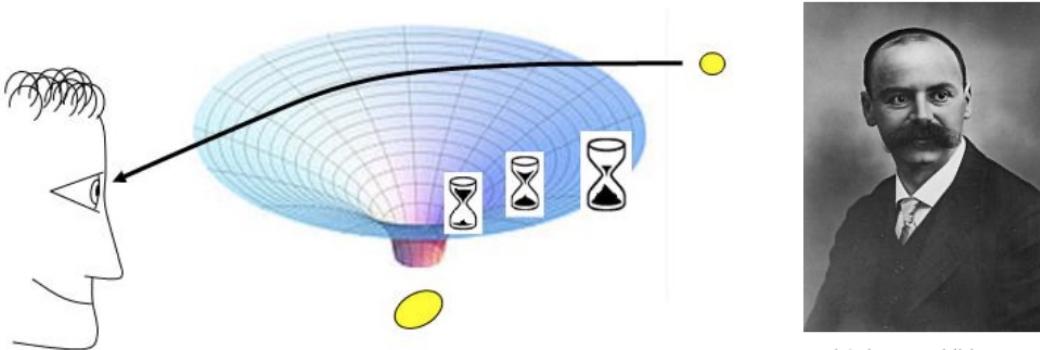


Figure 13: Descriptive representation of the Schwarzschild metric: A two-dimensional area of space is considered. In the vicinity of the mass, the distances are lengthened. This is illustrated by the fact that the surface has been stretched in a visually existing further direction. Time slows down near the mass. This is illustrated by the fact that relatively little sand has trickled through the hourglass. The ray of light entering the eye from the light source would have to traverse a very stretched surface if it were to travel closer to the mass. Therefore, the light beam makes an arc around the mass, to quote CARMESIN [3].

**Exercise 137.** In Einstein's theory of General Relativity, the SCHWARZSCHILD metric (also known as *the Schwarzschild solution*) is the first solution of Einstein's field equations, which was found in 1916 by the German astrophysicist Karl Schwarzschild (1873 - 1916). It describes the gravitational field of a homogeneous, non-charged and non-rotating sphere and is an example of the curvature of space-time, where light paths and trajectories follow the curvature of the space-time metric.

The Schwarzschild metric is a spherically symmetric Lorentzian metric defined on the subset  $U = \mathbb{R} \times (2M, \infty) \times S^2$ <sup>15</sup> in ambient 4D spacetime.

In Schwarzschild coordinates  $(t, r, \theta, \phi) \in U$  the SCHWARZSCHILD  $1^{st}$  fundamental form

<sup>15</sup>where  $S^2 \subset \mathbb{R}^3$  is the unit sphere in  $\mathbb{R}^3$ .

has the expression<sup>16</sup> with sign convention  $(-, +, +, +)$

$$ds^2 = - \left(1 - \frac{2M}{r}\right) dt^2 + \frac{1}{1 - \frac{2M}{r}} dr^2 + r^2 d\theta^2 + r^2 \sin^2 \theta d\phi^2 \quad (9.7)$$

resp. the SCHWARZSCHILD metric tensor is the matrix  $g$  with  $h(r) := 1 - \frac{2M}{r}$  and sign convention  $(+, -, -, -)$

$$g = \begin{bmatrix} h(r) & 0 & 0 & 0 \\ 0 & -\frac{1}{h(r)} & 0 & 0 \\ 0 & 0 & -r^2 & 0 \\ 0 & 0 & 0 & -r^2 \sin(\theta)^2 \end{bmatrix} \quad (9.8)$$

a. Verify using the arbitrary  $h(r)$  term:

$$R = \frac{4 \frac{\partial h(r)}{\partial r}}{r} - \frac{2}{r^2} + \frac{2h(r)}{r^2} + \frac{\partial^2 h(r)}{\partial r \partial r}$$

*Solution.* We use wxMAXIMA. First we define and compute the RICCI curvature tensor  $\text{Ric}$  with package `ctensor` and the RICCI curvature scalar  $R$ . We specify the 4-dimensional `ct(ensor)-coord(inate)s` in (0) and declare the explicit dependence of function  $h(r)$  of  $r$  in (1). The rest of the code is standard now.

```
/* MAXIMA - DiffGeo24b.txt */
"RICCI curvature scalar for the SCHWARZSCHILD metric";
load(ctensor)$
ct_coords : [t,r,theta, phi]$ /* (0) */
depends(h,r); /* (1) */
dim : 4$
lg : zeromatrix(4,4)$
lg[1,1] : h(r)$
lg[2,2] : - 1/h(r)$
lg[3,3] : - r^2$ 
lg[4,4] : - r^2*sin(theta)^2$
cmetric(false)$

"RICCI tensor";
ricci(true); /* compute RICCI tensor */
cdisplay(ric); /* show RICCI tensor */

"RICCI scalar curvature R";
R : scurvature(); /* (2) */
```

---

<sup>16</sup>By replacing  $M$  with  $GM/c^2$ , with  $G$  as the gravitational constant and  $t$  with  $c \cdot t$  ( $c$  speed of light), one reconnects to the physical system of measurements.

MAXIMA output:

$$\begin{aligned} (\%o18) \quad & \text{RICCI scalar curvature } R \\ (\text{R}) \quad & \frac{r^2 \left( \frac{d^2}{dr^2} h(r) \right) + 4 r \left( \frac{d}{dr} h(r) \right) + 2 h(r) - 2}{r^2} \end{aligned}$$

▷ Paste the code in MAXIMA *on line*

- b. Verify: if  $h(r) := 1 - 2\frac{M}{r}$  with  $M \in \mathbb{R}_+$  arbitrary, then  $R = 0$ .
- c. Do solution a. using our user-defined functions, i.e. do not use `ctensor`.

**Remark.** 1. In this example MAXIMA's `ctensor` package works in a 4D space with coordinates  $(t, r, \theta, \phi)$ , but using tensor techniques to calculate the CHRISTOFFEL tensors, the RIEMANN curvature tensors or the RICCI curvature scalar gives the expected results.  
 2. We use the conventions and notations in [23, p.5, p.52, p.96].

**Exercise 138.** (Post-EINSTEIN-SCHWARZSCHILD-Metric)

- a. Read the text ▷ Post-Einstein-Schwarzschild-Metrik . Although it is written in German, you should be able to recognize and understand all formulas.
- b. Using the metric tensor in (3) on p.2 of this script, calculate the CHRISTOFFEL tensors, the RIEMANN curvature tensors or the RICCI curvature scalar using tensor techniques by MAXIMA.

## 9.6 The RICCI scalar of the GÖDEL Spacetime Metric

REVIEWS OF MODERN PHYSICS

VOLUME 21, NUMBER 3

JULY, 1949

# An Example of a New Type of Cosmological Solutions of Einstein's Field Equations of Gravitation

KURT GÖDEL  
*Institute for Advanced Study, Princeton, New Jersey*

448

KURT GÖDEL

where  $a$  is a positive number. The matrices of the  $g_{ik}$  and  $g^{ik}$ , therefore, are the two:

$$a^2 \cdot \begin{vmatrix} 1 & 0 & e^{x_1} & 0 \\ 0 & -1 & 0 & 0 \\ e^{x_1} & 0 & e^{2x_1}/2 & 0 \\ 0 & 0 & 0 & -1 \end{vmatrix}, \quad \frac{1}{a^2} \cdot \begin{vmatrix} -1 & 0 & 2e^{-x_1} & 0 \\ 0 & -1 & 0 & 0 \\ 2e^{-x_1} & 0 & -2e^{-2x_1} & 0 \\ 0 & 0 & 0 & -1 \end{vmatrix}.$$

Owing to the fact that only two of the forty  $\partial g_{ik}/\partial x_i$  are  $\neq 0$ , namely  $\partial g_{22}/\partial x_1$  and  $\partial g_{02}/\partial x_1$ , the  $\Gamma_{i,kl}$  and  $\Gamma_{kl}^i$  can very easily be computed. One obtains the values:

$$\begin{aligned}\Gamma_{0,12} &= -\Gamma_{1,02} = \Gamma_{2,01} = (a^2/2)e^{x_1}, \\ \Gamma_{1,22} &= -\Gamma_{2,12} = -(a^2/2)e^{2x_1}, \\ \Gamma_{01}{}^0 &= 1, \quad \Gamma_{12}{}^0 = \Gamma_{02}{}^1 = e^{x_1}/2, \\ \Gamma_{22}{}^1 &= e^{2x_1}/2, \quad \Gamma_{01}{}^2 = -e^{-x_1}.\end{aligned}$$

These  $\Gamma_{i,kl}$  and  $\Gamma_{kl}^i$ , and those obtained from them by interchanging the last two (or the lower two) indices are the only ones that do not vanish.

Using for  $R_{ik}$  the formula<sup>7</sup>

$$R_{ik} = \frac{\partial}{\partial x_\sigma} \Gamma_{ik}{}^\sigma - \frac{1}{2} \frac{\partial^2 \log g}{\partial x_i \partial x_k} + \frac{1}{2} \Gamma_{ik}{}^\sigma \frac{\partial \log g}{\partial x_\sigma} - \Gamma_{\sigma i}{}^\rho \Gamma_{\rho k}{}^\sigma,$$

and taking account of the fact that  $\partial/\partial x_i$ , except for  $i=1$ , vanishes for every magnitude of the solution, and that  $g = (a^8/2)e^{2x_1}$ , we obtain

$$R_{ik} = \frac{\partial}{\partial x_1} \Gamma_{ik}{}^1 + \Gamma_{ik}{}^1 - \Gamma_{\rho i}{}^\sigma \Gamma_{\sigma k}{}^\rho.$$

This yields the values for the  $R_{ik}$

$$R_{00}=1, \quad R_{22}=e^{2x_1}, \quad R_{02}=R_{20}=e^{x_1};$$

all other  $R_{ik}$  vanish. Hence,

$$R=1/a^2,$$

**Exercise 139.** For the given metric tensor  $(g_{ik})$ , check the results of GöDEL's calculations for the CHRISTOFFEL tensors, the RICCI tensor and the RICCI curvature scalar  $R = \frac{1}{2}$ .

are satisfied

The sign of  
site of the  
correspondence

### 3. PROO

That the spaces obtained from the vector field exist a system on the vectors the skew

$$a_{ikl} = v_i$$

should vary  
responding



however,  
 $0, 0, 0, \sqrt{2}$   
 over, that  
 $x_0$ -lines in

If  $v$  is matter, the thogonal in a local rest at the follows at

The pro  
verifiable

## 9.7 Exercises

Let's calculate the RICCI curvature tensor  $\text{Ric}$  and the RICCI curvature scalar  $R$  for our small collection of classical 2D surfaces. Sometimes you should help MAXIMA to simplify trigonometric expressions or to simplify fractions.

- Calculate the RICCI curvature tensor  $\text{Ric}$  and RICCI curvature scalar  $R$  for the following surfaces using MAXIMA ...

**Exercise 140.** ... the saddle  $X(u, v) = (u, v, uv)$ .

**Exercise 141.** the cylinder  $X(u, v) = (a \cos(v), a \sin(v), u)$ . Choose  $a = 1$ .

**Exercise 142.** the cone  $X(u, v) = (u \cos(v), u \sin(v), u)$ .

**Exercise 143.** the elliptic paraboloid  $X(u, v) = (u, v, u^2 + v^2)$ .

**Exercise 144.** the hyperbolic paraboloid  $X(u, v) = (u, v, u^2 - v^2)$ .

**Exercise 145.** the monkey saddle  $X(u, v) = (u, v, u^3 - 3uv^2)$ .

**Exercise 146.** the unit sphere  $X(u, v) = (\cos(u) \sin(v), \sin(u) \sin(v), \cos(v))$ .

**Exercise 147.** the catenoid  $X(u, v) = (c \cosh \frac{v}{c} \cos u, c \cosh \frac{v}{c} \sin u, v)$ . Choose  $c = 1$ .

**Exercise 148.** the torus  $X(u, v) = ((b + a \cos(v)) \cos(u), (b + a \cos(v)) \sin(u), a \sin(v))$ .

**Exercise 149.** the function graph  $X(u, v) = (u, v, f(u, v))$ .

**Exercise 150.** the surface of revolution  $X(u, v) = (f(u) \cos(v), f(u) \sin(v), g(u))$ .

**Exercise 151.** the PLÜCKER conoid  $X(u, v) = (v \cdot \cos(u), v \cdot \sin(u), 2 \cos(u) \cdot \sin(u))$ .

**Exercise 152.** the SCHERK surface  $X(u, v) = (u, v, \ln(\frac{\cos v}{\cos u}))$ .

**Exercise 153.** the ENNEPER surface  $X(u, v) = (u - \frac{1}{3}u^3 + uv^2, v - \frac{1}{3}v^2 + vu^2, u^2 - v^2)$ .

**Exercise 154.** the BELTRAMI pseudosphere

$$X(u, v) = (\sin u \cos v, \sin u \sin v, \cos u + \ln(\sin u(\cos u + 1)))$$

**Exercise 155.** the hexenhut  $X(u, v) = (\alpha/\sqrt{u} \cdot \cos v, \alpha/\sqrt{u} \cdot \sin v, u)$  where  $\alpha^2 = \frac{2}{3\sqrt{3}}$ .

**Exercise 156.** the 'hexe' surface  $X(u, v) = (\sqrt{u} \cdot \cos v, \sqrt{u} \cdot \sin v, u)$ .

**Exercise 157.** the BIANCHI surface

$$X(u, v) = \begin{bmatrix} \frac{2 \cdot \sqrt{v^2+1} \cdot \sin(u) \cdot \cos(-v+\arctan(v))}{1+v^2 \sin(u)^2} \\ \frac{-2 \cdot \sqrt{v^2+1} \cdot \sin(u) \cdot \sin(-v+\arctan(v))}{1+v^2 \sin(u)^2} \\ \ln(\tan(\frac{1}{2}u)) + \frac{2 \cdot \cos(u)}{1+v^2 \sin(u)^2} \end{bmatrix}$$

## 10 The Fundamental Theorem

First we quote the well known equations from WEINGARTEN, GAUSS and CODAZZI-MAINARDI. Then we formulate the legendary *Fundamental Theorem of surface theory*.

### 10.1 The WEINGARTEN Equations

The WEINGARTEN equations express the partial derivatives of the surface normal  $\vec{n}$  as a linear combination of the tangential basis vectors  $X_u$  and  $X_v$ , where its coefficient are calculated only using the coefficients of the 1<sup>st</sup> and 2<sup>nd</sup> fundamental forms. These equation were found by Weingarten in 1861.

#### Theorem. (WEINGARTEN Equations)

LET  $X: U \rightarrow S \subset \mathbb{R}^3$  be a parameterization of a surface  $S$  in  $\mathbb{R}^3$ .

Let  $(X_u, X_v) := (\frac{\partial X}{\partial u}, \frac{\partial X}{\partial v})$  be the local basis in  $T_p S$ .

Let  $g = (g_{ij}) = \begin{bmatrix} E & F \\ F & G \end{bmatrix}$  be the surface metric tensor of  $X$ .

Let  $h = (h_{ij}) = \begin{bmatrix} L & M \\ M & N \end{bmatrix}$  be the 2<sup>nd</sup> fundamental tensor of  $X$ .

Let  $\vec{n} := \frac{X_u \times X_v}{|X_u \times X_v|}$  be the unit normal of  $S$  in  $p$ .<sup>17</sup>

THEN we have the two WEINGARTEN equations

$$\vec{n}_u = \frac{MF - LG}{EG - F^2} \cdot X_u + \frac{LF - ME}{EG - F^2} \cdot X_v \quad (10.1)$$

$$\vec{n}_v = \frac{NF - MG}{EG - F^2} \cdot X_u + \frac{MF - NE}{EG - F^2} \cdot X_v \quad (10.2)$$

**Remark.** For a proof cf. [13, p.26].

**Example 19.** (Check the WEINGARTEN Equations for the helicoid)

We consider the parametric surface helicoid with shape constant  $c$ .

To verify the validity of the WEINGARTEN Equations we use the surface basis  $X_u, X_v$ .

```
/* MAXIMA - DiffGeo25.txt */
"WEINGARTEN equations for the helicoid";
define( X(u,v), [u*cos(v), u*sin(v), c* v] )$ 
U : [u,v]$           /* surface coordinates */
"metric tensor of helicoid:";
g : matrix( [1, 0], [0, c^2+u^2] )$ 
E: g[1,1]; F: g[1,2]; G: g[2,2];           /*(1)*/
                                         /*(1)*/

"known 2nd fundamental tensor";
h : matrix( [0, -c/sqrt(c^2+u^2)], [-c/sqrt(c^2+u^2),0]);
```

<sup>17</sup>To avoid name clash, we use the symbol  $\vec{n}$  for the normal vector, because  $N$  is used in the formulae to denote the 4th coefficient of the sffh.

```

L:h[1,1]; M:h[1,2]; N:h[2,2];                                /*(2)*/

define( cross(v,w), [v[2]*w[3] - w[2]*v[3],
                    w[1]*v[3] - v[1]*w[3],
                    v[1]*w[2] - w[1]*v[2]]);

"surface unit normal";
Xu : diff(X(u,v),u);
Xv : diff(X(u,v),v);
n : cross(Xu, Xv);           /*(3)*/
n : trigsimp(n);

mag(v) := sqrt(v.v);      /* magnitude of vector v */
No   : n/trigsimp(mag(n)); /* unit normal vector */ /*(5)*/

"compare LHS and RHS of WEINGARTEN eq. (10.1)";
RHS101 : 1/(E*G - F^2)*(M*F - L*G)*Xu +
          1/(E*G - F^2)*(L*F - M*E)*Xv;
RHS101;

LHS101 : diff(No,u);
LHS101;

```

▷ Paste the code in MAXIMA<sup>on line</sup> or ▷ SAGEMATHCELL

MAXIMA output:

$$\left| \begin{array}{l}
 \text{(RHS101)} \left[ -\frac{u \sin(v)}{(u^2+1)^{3/2}}, \frac{u \cos(v)}{(u^2+1)^{3/2}}, \frac{1}{(u^2+1)^{3/2}} \right] \\
 \text{(LHS101)} \left[ -\frac{u \sin(v)}{(u^2+1)^{3/2}}, \frac{u \cos(v)}{(u^2+1)^{3/2}}, \frac{1}{\sqrt{u^2+1}} - \frac{u^2}{(u^2+1)^{3/2}} \right] \\
 \text{(%o35)} \quad \left[ -\frac{u \sin(v)}{(u^2+1)^{3/2}}, \frac{u \cos(v)}{(u^2+1)^{3/2}}, \frac{\sqrt{u^2+1}}{u^4+2 u^2+1} \right]
 \end{array} \right|$$

**Comment.** The 1<sup>st</sup> and 2<sup>nd</sup> ff  $g$  and  $h$  are known and given here directly in their matrices. In (1) we give the coefficients of the fff  $g$  their classic names  $E, F, G$  and in (2) we name the coefficients of the sff  $h$  classic as  $L, M, N$ . Line (3) constructs and simplifies (by mind) the normal vector  $n$  by means of the cross product, calculates his length) and normalize  $n$  to unit length (5).

We then write down the 1<sup>st</sup> Weingarten equation (10.1) separated in the LHS term named LHS101 and the RHS term abbreviated RHS101 in MAXIMA. We have to verify, that RHS101=LHS101 in order to show the equality. The RHS is delivered automatically by MAXIMAs output, to check that this term is equal to the LHS term we help MAXIMA a little bit with ratsimp. – q.e.d.

**Exercise 158.** Verify the validity of the WEINGARTEN equation (10.2) for the helicoid.

**Exercise 159.** Verify the validity of the WEINGARTEN equations for the surface  $z = uv$ .

**Exercise 160.** Verify the validity of the WEINGARTEN equations for the monkey saddle, the cone and the hyperbolic paraboloid and ... what you like.

## 10.2 The GAUSS Equations

The Gauss equations are a system of three partial differential equations. They describe a connection between the tangential vectors, the unit normal vector of a parametrized surface and their coefficients of the 2<sup>nd</sup> fundamental form.

### Theorem. (GAUSS Equations)

LET  $X: U \rightarrow S \subset \mathbb{R}^3$  be a parameterization of a surface  $S$  in  $\mathbb{R}^3$ .

Let  $E = (E_1, E_2) := (X_u, X_v) = (\frac{\partial X}{\partial u}, \frac{\partial X}{\partial v})$  be the local base in  $T_p S$ .  $X_{uu} := \frac{\partial^2 X}{\partial u^2}$  etc.

Let  $g = (g_{ij})$  be the surface metric tensor of  $X$ .

Let  $h = (h_{ij}) = \begin{bmatrix} L & M \\ M & N \end{bmatrix}$  be the 2<sup>nd</sup> fundamental tensor of  $X$ .

Let  $\vec{n} := \frac{X_u \times X_v}{|X_u \times X_v|}$  be the unit normal of  $S$  in  $p$ .<sup>18</sup>

Let  $\Gamma^{II} = (\Gamma^1, \Gamma^2)$  be the CHRISTOFFEL symbol coefficients of 2<sup>nd</sup> kind.

THEN we have the three GAUSS equations

$$X_{uu} = \Gamma_{11}^1 \cdot X_u + \Gamma_{11}^2 \cdot X_v + L \cdot \vec{n} \quad (10.3)$$

$$X_{uv} = \Gamma_{12}^1 \cdot X_u + \Gamma_{12}^2 \cdot X_v + M \cdot \vec{n} \quad (10.4)$$

$$X_{vv} = \Gamma_{22}^1 \cdot X_u + \Gamma_{22}^2 \cdot X_v + N \cdot \vec{n} \quad (10.5)$$

*Proof.* cf. OLOFF [23, p.93], PRESSLEY [24, p.240] or ▷ Derivation of classical equations. □

**Remark.** If we denote the coordinate basis by  $E = (E_1, E_2)$  and the parameters as  $(u^1, u^2) = (u, v)$  with integer indices, then the GAUSS equations may be written by

$$\begin{aligned} \frac{\partial E_1}{\partial u^1} &= \Gamma_{11}^1 \cdot E_1 + \Gamma_{11}^2 \cdot E_2 + L \cdot \vec{n} \\ \frac{\partial E_1}{\partial u^2} &= \Gamma_{12}^1 \cdot E_1 + \Gamma_{12}^2 \cdot E_2 + M \cdot \vec{n} \\ \frac{\partial E_2}{\partial u^2} &= \Gamma_{22}^1 \cdot E_1 + \Gamma_{22}^2 \cdot E_2 + N \cdot \vec{n} \end{aligned}$$

---

<sup>18</sup>To avoid name clash, we use the symbol  $\vec{n}$  for the normal vector, because  $N$  is used in the formulae to denote the 4th coefficient of the sff.

and then compactly expressed with the EINSTEIN summation convention by one tensorial equation ( $i, j \in \{1, 2\}$ )

$$\frac{\partial E_i}{\partial u^j} = \Gamma_{ij}^k \cdot E_k + h_{ij} \cdot \vec{n} \quad (10.6)$$

We now verify the validity of the Gauss Equations for some surfaces.

**Example 20.** (Gauss Equations for the helicoid)

We consider the parametric surface helicoid with shape constant  $c$ . To verify the validity of the GAUSS Equations we use the surface basis  $X_u, X_v$  and their derivatives  $X_{uu}, X_{uv} = X_{vu}, X_{vv}$ , i.e. the second partial derivatives of the tangential basis vectors, which appears on the LHS of the GAUSS equations.

```
/* MAXIMA - DiffGeo26.txt */
"GAUSS equations for the helicoid";
c :1;
define( X(u,v), [u*cos(v), u*sin(v), c* v] )$ 
U : [u,v]$           /* surface coordinates */
"metric tensor of helicoid:";

g : matrix( [1, 0], [0, c^2+u^2] )$ 
E: g[1,1]; F: g[1,2]; G: g[2,2];

"known 2nd fundamental tensor";
h : matrix( [0, -c/sqrt(c^2+u^2)], [-c/sqrt(c^2+u^2),0]);
L:h[1,1]; M:h[1,2]; N:h[2,2];          /*(1)*/

gu : invert(g);
gu : trigsimp(gu);

Xu : diff(X(u,v),u);
Xv : diff(X(u,v),v);
Xuu : diff(X(u,v),u,2);      /* 2nd partial derivative w.r.t. u */
Xuv : diff( diff(X(u,v),u), v);
Xvv : diff(X(u,v),v,2);

"CHRISTOFFEL symbol of 2nd kind"$
ChrII(k,i,j) := 1/2* sum( gu[k,m]*( diff(g[m,i],U[j]) +
diff(g[m,j],U[i]) - diff(g[i,j],U[m]) ), m,1,2)$

define( cross(v,w), [v[2]*w[3] - w[2]*v[3],
                     w[1]*v[3] - v[1]*w[3],
                     v[1]*w[2] - w[1]*v[2]]);
```

```

"surface unit normal";
n : cross(Xu, Xv);           /*(2)*/
n : trigsimp(n);
mag(v) := sqrt(v.v);         /* magnitude of vector v */
No   : n/trigsimp(mag(n));   /*(3)*/

"GAUSS equation check";

Xuu - (ChrII(1,1,1)*Xu + ChrII(2,1,1)*Xv + L*No);      /*(G1)*/
Xuv - (ChrII(1,1,2)*Xu + ChrII(2,1,2)*Xv + M*No);      /*(G2)*/
ratsimp(%);                                                 /*(G2a)*/
Xvv - (ChrII(1,2,2)*Xu + ChrII(2,2,2)*Xv + N*No);      /*(G3)*/

```

MAXIMA output:

$\text{(%)30} \quad \text{GAUSS equation check}$ $\text{(%)31} \quad [0, 0, 0]$ $\text{(%)32} \quad \left[ \frac{u^2 \sin(v)}{u^2 + 1} + \frac{\sin(v)}{u^2 + 1} - \sin(v), -\frac{u^2 \cos(v)}{u^2 + 1} - \frac{\cos(v)}{u^2 + 1} + \cos(v), 0 \right]$ $\text{(%)33} \quad [0, 0, 0]$ $\text{(%)34} \quad [0, 0, 0]$
--

▷ Paste the code in MAXIMA *on line* or ▷ SAGEMATHCELL

**Comment.** The 1<sup>st</sup> and 2<sup>nd</sup> ff  $g$  and  $h$  are known and given here directly in their matrices. In (1) we give the coefficients of the sff their usual names  $L, M, N$ ). Line (2) constructs and then simplifies the normal vector  $n$  by means of the cross product, calculates his length and normalize  $n$  to unit length  $N_o$  in (3).

(G1), (G2),(G3) are the three GAUSS equations expressed in MAXIMA. All terms of the three equations are collected on the LHS, so we have to verify, that the result is 0 in order to show the validity. The result  $(G1) = 0$  and  $(G3) = 0$  is delivered automatically by MAXIMAS output, to check that  $(G2) = 0$  we help MAXIMA a little bit in (G2a). – q.e.d.

**Exercise 161.** Verify the validity of the GAUSS equations for the surface  $z = uv$ .

**Exercise 162.** Verify the validity of the GAUSS Equations for

- the monkey saddle
- the cone
- the hyperbolic paraboloid

...

### 10.3 The CODAZZI-MAINARDI Equations

The CODAZZI-MAINARDI equations are important formulas which link together the metric tensor  $g$  and the second fundamental form  $h$  of a parametric surface by means of the CHRISTOFFEL symbol of the 2<sup>nd</sup> kind..

**Theorem. (CODAZZI-MAINARDI Equations)**

LET  $X: U \rightarrow S \subset \mathbb{R}^3$  be a parameterization of a surface  $S$  in  $\mathbb{R}^3$ .

Let  $g = (g_{ij})$ ,  $h = (h_{ij}) = \begin{bmatrix} L & M \\ M & N \end{bmatrix}$  be the 1<sup>st</sup> and 2<sup>nd</sup> fundamental tensors of  $X$  .

Let  $\Gamma^{II} = (\Gamma^1, \Gamma^2)$  be the CHRISTOFFEL symbol coefficients of 2<sup>nd</sup> kind as above.

THEN we have the two CODAZZI-MAINARDI equations

$$L_v - M_u = L \cdot \Gamma^1_{12} + M \cdot (\Gamma^2_{12} - \Gamma^1_{11}) - N \cdot \Gamma^2_{11} \quad (10.7)$$

$$M_v - N_u = L \cdot \Gamma^1_{22} + M \cdot (\Gamma^2_{22} - \Gamma^1_{12}) - N \cdot \Gamma^2_{12} \quad (10.8)$$

or by compacting both equations via the EINSTEIN summation convention into one tensorial equation for  $i, j, l \in \{1, 2\}$

$$\frac{\partial h_{ij}}{\partial u^l} - \Gamma^k_{il} \cdot h_{kj} = \frac{\partial h_{il}}{\partial u^j} - \Gamma^k_{ij} \cdot h_{kl} \quad (10.9)$$

*Proof.* cf. PRESSLEY [24, p.241], BANCHOFF [2, p.229] or ▷ Derivation of classical equations. □

**Example 21.** (CODAZZI-MAINARDI equations for the helicoid)

We consider the parametric surface helicoid with shape constant  $c$ . To verify the validity of the CODAZZI-MAINARDI equations we need the coefficients  $L, M, N$  and the partial derivatives of these coefficients of the sff  $h$  w.r.t.  $u$  and  $v$ , i.e.  $L_v, M_u, M_v, N_u$  and the coefficients of  $\Gamma^{II}$ . Again we collect all terms of the equation on the LHS and show, that LHS = 0.

```
/* MAXIMA - DiffGeo27.txt */
"CODAZZI-MAINARDI equations for the helicoid";

c :1;
define( X(u,v), [u*cos(v), u*sin(v), c* v] )$ 
U : [u,v]$           /* surface coordinates */
"metric tensor of helicoid:";
g : matrix( [1, 0], [0, c^2+u^2] )$ 
E: g[1,1]; F: g[1,2]; G: g[1,1];

"known 2nd fundamental tensor";
h : matrix( [0, -c/sqrt(c^2+u^2)], [-c/sqrt(c^2+u^2),0]);
```

```

L:h[1,1]; M:h[1,2]; N:h[2,2];

gu : invert(g);
gu : trigsimp(gu);

Xu : diff(X(u,v),u);
Xv : diff(X(u,v),v);

Xuu : diff(X(u,v),u,2);      /* 2nd partial derivative w.r.t. u */
Xuv : diff( diff(X(u,v),u), v);
Xvv : diff(X(u,v),v,2);

"CHRISTOFFEL symbol of 2nd kind"
ChrII(k,i,j) := 1/2* sum( gu[k,m]*( diff(g[m,i],U[j]) +
                           diff(g[m,j],U[i]) - diff(g[i,j],U[m]) ), m,1,2)$

"CODAZZI equation check";
diff(L,v)-diff(M,u) - (L*ChrII(1,1,2) +
M*(ChrII(2,1,2)-ChrII(1,1,1)) - N*ChrII(2,1,1));
diff(M,v)-diff(N,u) - (L*ChrII(1,2,2) -
M*(ChrII(2,2,2)-ChrII(1,1,2)) - N*ChrII(2,1,2));

```

▷ Paste the code in MAXIMA<sup>on line</sup> or ▷ SAGEMATHCELL

MAXIMA output:

CODAZZI equation check

0

0

**Comment.** We implement the CODAZZI-MAINARDI equations 1:1 from equations (10.9) and (10.10) in lines (CM1) and (CM2). MAXIMAs output displays that  $(CM1) = (CM2) = 0$ . Ok.

**Exercise 163.** Verify the validity of the CODAZZI-MAINARDI equations for the surface  $z = uv$ .

**Exercise 164.** Verify the validity of the CODAZZI-MAINARDI equations for

- the monkey saddle
- the cone
- the hyperbolic paraboloid
- a function graph
- ...

**Exercise 165.** Verify the validity of the CODAZZI-MAINARDI equations for the surface of revolution with the parametrization  $X: U \rightarrow \mathbb{R}^3$  defined by  $X(u, v) = (u \cos(v), u \sin(v), f(u))$  with  $f: \mathbb{R} \rightarrow \mathbb{R}_+$  arbitrary.

Why is it important to test if the CODAZZI-MAINARDI equations are fulfilled for the 1<sup>st</sup> and 2<sup>nd</sup> fundamental forms of a surface? The answer is given in the so-called *Fundamental Theorem of Surfaces Theory*.

## 10.4 The Fundamental Theorem of Surfaces Theory

The question arises to what extent a parametric surface is (uniquely) determined by specifying the first and second fundamental forms. If one calculates the second derivatives of the basis vectors of the tangent space of the surface, one finds that the coefficients of the first and second fundamental forms cannot be chosen completely independently of each other. The necessary conditions work in the form of the CODAZZI-MAINARDI equations and BRIOSCHI's formula. It is a fact that these necessary conditions are also sufficient. Looking back, this motivates our short treatment of the CODAZZI-MAINARDI equation.

With the CODAZZI-MAINARDI equations we are now ready to formulate the so-called *fundamental theorem of the surface theory*, which is analogous to the fundamental theorem of curve theory. It was published by O. BONNET in 1867.

**Theorem. (The Fundamental Theorem of Surfaces Theory)**

LET the coefficients of the matrix  $g, h: U \rightarrow \mathbb{R}$  of the first and second fundamental forms

- (1) be  $C^2$ -functions of the parameters  $u$  and  $v$  and
- (2) satisfy the CODAZZI-MAINARDI equations and
- (3) satisfy BRIOSCHI's formula for the calculation of the GAUSS curvature  $K$ ,

THEN *there exists a uniquely determined (up to translations and rotations) surface  $S$ , which owns these given first and second fundamental forms, i.e.  $I_S = g$  and  $II_S = h$ .*

*Proof.* Cf. BANCHOFF [2, p.238], LIPSCHUTZ [20, p.203] or ▷ FuTheorem of surfaces, GLUCK [10, p.34] □

**Remark.** The conditions (2) and (3) are called the *compatibility equations*. They answer the important existence question, if it is possible to construct a 2D surface with prescribed 1<sup>st</sup> and 2<sup>nd</sup> fundamental tensor coefficients.

## 10.5 Exercises

**Exercise 166.** Verify that the functions  $E = 1 + 4u^2, F = -4uv, G = 1 + 4v^2, L = 2(4u^2 + 4v^2 + 1)^{-1/2}, M = 0, N = -2(4u^2 + 4v^2 + 1)^{-1/2}$  satisfy the compatibility equations.

**Exercise 167.** (existence of a surface with prescribed fundamental coefficients)

Let  $(E, F, G) = (1, 0, \sin^2 u)$  and  $(L, M, N) = (1, 0, \sin^2 u)$  be candidates for the coefficients for the 1<sup>st</sup> and 2<sup>nd</sup> fundamental tensors of a surface  $S$ , cf. [20, p.204].

- a. Check, that  $E, \dots, N$  satisfy the compatibility equations, i.e. that the CODAZZI-MAINARDI equations and BRIOSCHI's formula are fulfilled.
- b. Show, that the surface  $S$  with this set of given fundamental coefficients is the sphere of radius 1.

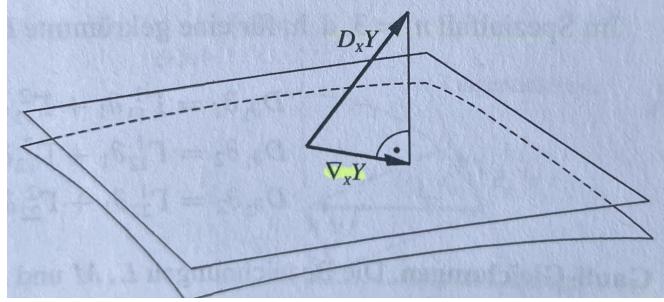
**Exercise 168.** Study the example 10.2 in [24, p.242] giving a cylinder of radius 1.

**Exercise 169.** Verify that there is no surface  $S$  parametrized by  $X(u, v)$  with  $1^{st}$  and  $2^{nd}$  fundamental forms  $I = du^2 + \cos^2 u dv^2$  and  $II = \cos^2 u du^2 + dv^2$ , cf. [24, p.244].

**Exercise 170.** Do problem 9 in GLUCK [10, p.37].

**Exercise 171.** Do problem 10 in GLUCK [10, p.38].

## 11 Covariant Derivative



$D_X Y$ : the *directional derivative* in  $\mathbb{R}^3$ , e.g. grad.

The result is element of the *ambient* vector space.

Figure 14:  $\nabla_X Y$ : the *covariant derivative* in  $T_p S$ . The result is in the *tangential* vector space, i.e.  $\nabla_X Y$  is the orthogonal projection of the directional derivative  $D_X Y$  onto  $T_p M$ .

The CHRISTOFFEL symbol  $\Gamma$  will now be used to formulate the concept of the *covariant derivative* of two vector fields  $X$  and  $Y$  on the tangent space  $T_p S$  of a parametrized surface  $S$  in ambient space  $\mathbb{R}^3$ .

### 11.1 Definition

Let  $\phi: U \rightarrow S \subset \mathbb{R}^3$  be a parameterization of a surface  $S$  in  $\mathbb{R}^3$ .

Let  $E = (E_1, E_2) := (\phi_u, \phi_v)$  be the local base in  $T_p S$ .

Let  $g = (g_{ij})$  be the surface metric tensor of  $\phi$ .

Let  $X, Y: U \rightarrow T_p S$  be two vector fields defined by their coefficients as  $X = \sum_{i=1}^2 X^i \cdot E_i$  and  $Y = \sum_{i=1}^2 Y^i \cdot E_i$  in the local base  $E$ .

Let  $\Gamma^{II} = (\Gamma^1, \Gamma^2)$  be the CHRISTOFFEL tensor coefficients of 2<sup>nd</sup> kind.

Then we define – using EINSTEIN summation convention! – the *covariant derivation of Y w.r.t. X*, noted  $\nabla_X Y$  by

$$\nabla_X Y := X^i \frac{\partial Y^j}{\partial x^i} E_j + X^i Y^j \Gamma_{ij}^k E_k \quad (11.1)$$

or more explicit by decoding the EINSTEIN summation convention

$$\nabla_X Y := \left( X_u \frac{\partial Y^u}{\partial u} + X_v \frac{\partial Y^u}{\partial v} + \sum_{i,j=1}^2 X_i Y_j \Gamma_{ij}^u \right) E_u + \left( X_u \frac{\partial Y^v}{\partial u} + X_v \frac{\partial Y^v}{\partial v} + \sum_{i,j=1}^2 X_i Y_j \Gamma_{ij}^v \right) E_v \quad (11.2)$$

**Remark.** 1. Formula (11.1) is given e.g. in OLOFF [23, p.91] or THORPE [37, p.229].

2. Formula (11.2) is given in  $\triangleright$  Connexion de Levi-Civita.

3. In (11.2) the symbol  $\sum_{i,j=1}^2 \dots \equiv \sum_{i=1}^2 \sum_{j=1}^2 \dots$  means a double summation.

## 11.2 Implementation and Examples

In the following examples we calculate the two coefficients of the covariant derivative in (10.2) and then associate the results to  $E_1$  resp.  $E_2$ . We demonstrate this with three different surfaces and different vector fields.

**Example 22.** (covariant derivative of two vector fields on  $z = uv$ )

We consider the function surface  $f(u, v) = u \cdot v$  and its parametrization  $X := \phi$ .

```
/* MAXIMA - DiffGeo32.txt */
"Covariant derivation on surface z=uv";
define( X(u,v), [u, v, u*v] )$ 
U : [u,v]$                  /* surface coordinates */

"metric tensor of z=uv:";
g : zeromatrix(2,2);
g[1,1] : diff(X(u,v),u) . diff(X(u,v),u);
g[1,2] : diff(X(u,v),u) . diff(X(u,v),v);
g[2,1] : g[1,2];
g[2,2] : diff(X(u,v),v) . diff(X(u,v),v);
g;

gu : invert(g);
gu : ratsimp(gu);

"DEFINITION of CHRISTOFFEL symbol of 2nd kind"$
ChrII(k,i,j) := 1/2* sum( gu[k,m]*( diff(g[m,i],U[j]) +
                           diff(g[m,j],U[i]) - diff(g[i,j],U[m]) ), m,1,2)$

/* DEFINITION of covariant derivative via (10.2) */
covD(Y,X) := [ X[1]*diff(Y[1],u) + X[2]*diff(Y[1],v) +
                sum( sum( X[i]*Y[j]*ChrII(1,i,j), j,1,2 ), i,1,2 ) ,
                X[1]*diff(Y[2],u)+X[2]*diff(Y[2],v) +
                sum( sum( X[i]*Y[j]*ChrII(2,i,j), j,1,2 ), i,1,2 ) ];

V : [u,v^3] ;      /* i.e. ... = u *Eu + v^3 *Ev */ /* (1) */
W : [u*v, u+v];  /* i.e. ... = uv*Eu +(u+v)*Ev */
D1 : [1,0];
D2 : [0,1];

ratsimp(covD(W,V));
covD(D1,D1);
ratsimp( covD(D1,D2) );
covD(D2,D2);
```

- ▷ Paste the code in MAXIMA<sup>on line</sup> or ▷ SAGEMATHCELL

wxMAXIMA output:

```
(%o143) [0,0]
(%o144) [v/(v^2+u^2+1), u/(v^2+u^2+1)]
(%o145) [0,0]
```

**Comment.** The definition of the covariant derivative follows 1:1 the mathematical definition in (10.2), but we limit us to only calculate the pair of coefficients. In (1) we define four different vector fields and calculate their covariant derivatives.

**Example 23.** (covariant derivative of two vector fields on the helicoid)  
Consider the parametric surface of the helicoid.

```
/* MAXIMA - DiffGeo31.txt */
"Covariant derivation on the helicoid";
c : 1;
define( X(u,v), [u*cos(v), u*sin(v), c*v] )$ 
U : [u,v]$                      /* surface coordinates */

"metric tensor of helicoid";
g : zeromatrix(2,2);
g[1,1] : diff(X(u,v),u) . diff(X(u,v),u);
g[1,2] : diff(X(u,v),u) . diff(X(u,v),v);
g[2,1] : g[1,2];
g[2,2] : diff(X(u,v),v) . diff(X(u,v),v);
trigsimp(g[2,2]);
g;
gu : invert(g);
gu : trigsimp(gu);                  /* (1) */

"CHRISTOFFEL symbol of 2nd kind"
ChrII(k,i,j) := 1/2* sum( gu[k,m]*( diff(g[m,i],U[j]) +
                                diff(g[m,j],U[i]) - diff(g[i,j],U[m]) ), m,1,2)$

"DEFINITION: Covariant derivation of vector fields";
covD(Y,X) := [ X[1]*diff(Y[1],u) + X[2]*diff(Y[1],v) +
                sum( sum( X[i]*Y[j]*ChrII(1,i,j), j,1,2 ), i,1,2 ) ,
                X[1]*diff(Y[2],u)+X[2]*diff(Y[2],v) +
                sum( sum( X[i]*Y[j]*ChrII(2,i,j), j,1,2 ), i,1,2 ) ];

V : [u^2,v^2];      /* i.e. ... = u^2 *Eu + v^2 *Ev */
W : [u-v, u+v];    /* i.e. ... = (u-v)*Eu + (u+v)*Ev */
covD(W,V);
trigsimp(covD(W,V));                         /* (2) */
```

▷ Paste the code in MAXIMA *on line* or ▷ SAGEMATHCELL

MAXIMA output:

$$\text{(%)o22) } \left[ -u v^3 + (-u^2 - 1) v^2 + u^2, -\frac{u v^3 + (-2 u^2 - 1) v^2 - u^3 v - 2 u^4 - u^2}{u^2 + 1} \right]$$

**Comment.** To have a simplified output we help MAXIMA in (1) and (2) to simplify the coefficients of  $g$ ,  $gu$  and  $\text{covD}(W,V)$ . We use the well-known trigonometric Pythagoras formula via a call to `trigsimp` – this could also easily done by hand.

**Example 24.** (covariant derivative of two vector fields on the sphere)

We consider the parametric sphere of radius  $R$ . The aim of this example is to verify the covariant derivatives of the coordinate vector fields  $E_1 = E_u \equiv \partial_\theta$  and  $E_2 = E_v \equiv \partial_\varphi$ , whose results are given in OLOFF [23, p.92; p.95].

```
/* MAXIMA - DiffGeo33.txt */
"Covariant derivation on the sphere of radius R";
define( X(u,v), [R*sin(u)*cos(v), R*sin(u)*sin(v), R*cos(u)] )$ 
U : [u,v]$           /* surface coordinates */

"metric tensor of helicoid:";
g : zeromatrix(2,2);
  g[1,1] : diff(X(u,v),u) . diff(X(u,v),u);
  g[1,2] : diff(X(u,v),u) . diff(X(u,v),v);
  g[2,1] : g[1,2];
  g[2,2] : diff(X(u,v),v) . diff(X(u,v),v);
  trigsimp(g[2,2]);
g;

gu : invert(g);
gu : trigsimp(gu);

"DEFINITION of CHRISTOFFEL symbol of 2nd kind"$
ChrII(k,i,j) := 1/2* sum( gu[k,m]*( diff(g[m,i],U[j]) +
  diff(g[m,j],U[i]) - diff(g[i,j],U[m]) ), m,1,2)$

"DEFINITION of Covariant derivation of vector fields";
covD(W,V) := [ V[1]*diff(W[1],u) + V[2]*diff(W[1],v) +
  sum( sum( V[i]*W[j]*ChrII(1,i,j), j,1,2), i,1,2) , 
  V[1]*diff(W[2],u)+V[2]*diff(W[2],v) +
  sum( sum( V[i]*W[j]*ChrII(2,i,j), j,1,2), i,1,2) ];

trigsimp(covD(D1,D1));
trigsimp(covD(D1,D2));
trigsimp(covD(D2,D2));
```

- ▷ Paste the code in MAXIMA *on line* or ▷ SAGEMATHCELL

MAXIMA output:

(%o42)	[0, 0]
(%o43)	[0, $\frac{\cos(u)}{\sin(u)}$ ]
(%o44)	[- $\cos(u)\sin(u)$ , 0]

**Comment.** We interpret the MAXIMA output.

The output associated with line (o42) means

$$\begin{aligned} \text{covD}(D1, D1) &= \nabla_{D1} D1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \equiv 0 \cdot E_1 + 0 \cdot E_2 \\ &\equiv 0 \cdot \partial_\vartheta + 0 \cdot \partial_\varphi = \nabla_{\partial_\vartheta} \partial_\vartheta \Big|_{\text{OLOFF}} = 0 \end{aligned}$$

The output associated with line (o43) means

$$\begin{aligned} \text{covD}(D1, D2) &= \nabla_{D1} D2 = \begin{bmatrix} 0 \\ \cos(u)/\sin(u) \end{bmatrix} \\ &\equiv 0 \cdot E_1 + \frac{\cos(u)}{\sin(u)} \cdot E_2 \equiv 0 \cdot \partial_\vartheta + \cot(\varphi) \cdot \partial_\varphi \\ &= \nabla_{\partial_\vartheta} \partial_\varphi \Big|_{\text{OLOFF}} = \cot(\varphi) \cdot \partial_\varphi \end{aligned}$$

The output associated with line (o44) means

$$\begin{aligned} \text{covD}(D2, D2) &= \nabla_{D2} D2 = \begin{bmatrix} -\cos(u)\sin(u) \\ 0 \end{bmatrix} \equiv -\cos(u)\sin(u) \cdot E_1 + 0 \cdot E_2 \\ &\equiv -\cos(\vartheta)\sin(\vartheta) \cdot \partial_\vartheta + 0 \cdot \partial_\varphi = \nabla_{\partial_\vartheta} \partial_\varphi \Big|_{\text{OLOFF}} = -\cos(\vartheta)\sin(\vartheta) \cdot \partial_\vartheta \end{aligned}$$

These results for the covariant derivatives of the coordinate vector fields on the sphere agree with the results given by OLOFF [23, p.92; p.95].

### 11.3 Exercise

#### Exercise 172. (the directional derivative)

In the figure 14 we gave a first visual impression of what the difference of the covariant derivative of two vector fields and their directional derivative is.

OLOFF [23, p.88] gives the following definition of the *directional derivative*  $D_X Y$  of two vector fields on a parametrized surface  $\phi: U \rightarrow \mathbb{R}^3$  with  $\partial_1 \equiv \phi_u \equiv E_u, \partial_2 \equiv \phi_v \equiv E_v$ :

$$D_X Y := X^i \partial_i Y^k \partial_k \quad (11.3)$$

a. Interpret formula (10.3) using our  $E_u, E_v$  instead of  $\partial_1, \partial_2$ .

Be warned: (11.3) is decoded by means of the EINSTEIN summation convention .

- b. Implement  $D_X Y$  as function `dirD(Y,X)` in MAXIMA and calculate the directional derivatives of the vector fields of the forgoing examples.
- c. THORPE [37, p.55] gives an alternative geometric definition of the covariant derivative as the tangential component of the directional derivative, i.e. as orthogonal projection of the directional derivative onto the tangential space  $T_p S$  of the surface:

$$\nabla_X Y := D_X Y - (D_X Y \bullet N) \cdot N \quad (11.4)$$

where  $N$  is the unit normal vector (field) of the surface.

See also HITCHIN [14, p.64] or SHIFRIN [33, p.67]

Implement definition (11.4) in MAXIMA and reproduce some results of the examples of this chapter.

**Remark.** Definition (11.4) uses the normal vector and is therefore an extrinsic concept. The definition (11.2) uses only the 1<sup>st</sup> fundamental form and the CHRISTOFFEL tensor and belongs to intrinsic geometry.

*Consider the following parametric surfaces.*

*Task: Calculate the covariant derivatives  $\nabla_X Y$  of the vector fields  $X = (u^2, v^2) \equiv u^2 * E_u + v^2 * E_v$  and  $Y = (u - v, u + v) \equiv (u - v) * E_u + (u + v) * E_v$  on the respective surface  $X$  using MAXIMA ...*

**Exercise 173.** ... the cylinder  $X(u, v) = (a \cos(v), a \sin(v), u)$ . Choose  $a = 1$ .

**Exercise 174.** the cone  $X(u, v) = (u \cos(v), u \sin(v), u)$ .

**Exercise 175.** the hyperbolic paraboloid  $X(u, v) = (u, v, u^2 - v^2)$ .

**Exercise 176.** the monkey saddle  $X(u, v) = (u, v, u^3 - 3uv^2)$ .

**Exercise 177.** the function graph  $X(u, v) = (u, v, f(u, v))$ .

**Exercise 178.** the surface of revolution  $X(u, v) = (f(u) \cos(v), f(u) \sin(v), g(u))$ .

**Exercise 179.** the PLÜCKER conoid  $X(u, v) = (v \cdot \cos(u), v \cdot \sin(u), 2 \cos(u) \cdot \sin(u))$ .

**Exercise 180.** the SCHERK surface  $X(u, v) = (u, v, \ln(\frac{\cos v}{\cos u}))$ .

**Exercise 181.** the hexenhut  $X(u, v) = (\alpha/\sqrt{u} \cdot \cos v, \alpha/\sqrt{u} \cdot \sin v, u)$  where  $\alpha^2 = \frac{2}{3\sqrt{3}}$ .

**Exercise 182.** the 'hexe' surface  $X(u, v) = (\sqrt{u} \cdot \cos v, \sqrt{u} \cdot \sin v, u)$ .

**Exercise 183.** Do some more covariant derivatives of vector fields of your choice on surfaces you are interested in.



This ends our first steps into the Elementary Differential Geometry using MAXIMA as our companion.

## 12 Appendix: source code of diffgeoBox

This is a collection of the relevant definitions from the booklet. These functions should be used to do calculations without the package `ctensor` using MAXIMA.

```

##### (2023) Dr. W. G. Lindner, Leichlingen DE
#### diffgeoBox      Differential Geometry box
#####                 for calculations of curvature */

define( X(u,v), [u, v, u*v] )$ /* example surface parameterization */
U : [u,v]$                      /* surface coordinates */

"surface metric tensor"
gij(i,j) := diff(X(u,v),U[i]) . diff(X(u,v),U[j])$
g : matrix( [gij(1,1),gij(1,2)], [gij(2,1),gij(2,2)] );

"partial derivative w.r.t. u,v ";
Xu : diff(X(u,v),u);
Xv : diff(X(u,v),v);
E: Xu.Xu;   F: Xu.Xv;   G: Xv.Xv;

define( cross(v,w), [v[2]*w[3] - w[2]*v[3],
                    w[1]*v[3] - v[1]*w[3],
                    v[1]*w[2] - w[1]*v[2]]);

"calculate 2nd fundamental form h";
Xuu : diff(X(u,v),u,2);
Xuv : diff( diff(X(u,v),u), v);
Xvv : diff(X(u,v),v,2);

V : sqrt(determinant(g));

L : 1/V * cross(Xu, Xv) . Xu;
M : 1/V * cross(Xu, Xv) . Xuv;
N : 1/V * cross(Xu, Xv) . Xvv;
h : matrix( [L,M], [M,N] );

"shape operator A";
A : gu . h;

mag(v)  := sqrt(v.v);           /* magnitude of vector v */
trace(M) := M[1,1]+M[2,2];      /* mattrace is build-in */

"CHRISTOFFEL symbol Gamma_ijk of 1st kind";
ChrI(i,j,k) := diff( diff(X(u,v),U[i]), U[j]) . diff(X(u,v), U[k]);

```

```

"RIEMANN curvature tensor R_ijkl of 1st kind";
RI(b,m,s,q) := diff(ChrI(b,m,q),U[s]) - diff(ChrI(b,m,s),U[q]) +
               sum(ChrI(b,n,s)*ChrII(n,m,q),n,1,2) -
               sum( ChrI(b,n,q)*ChrII(n,m,s) ,n,1,2);

"CHRISTOFFEL symbol Gamma^k_ij of 2nd kind"$
ChrII(k,i,j) := 1/2* sum( gu[k,m]*( diff(g[m,i],U[j]) +
               diff(g[m,j],U[i]) - diff(g[i,j],U[m]) ), m,1,2)$

"RIEMANN curvature tensor R^i_jkl of 2nd kind"$
RII(b,m,s,q) := diff(ChrII(b,m,q),U[s]) - diff(ChrII(b,m,s),U[q]) +
               sum( ChrII(b,n,s)*ChrII(n,m,q), n,1,2) -
               sum( ChrII(b,n,q)*ChrII(n,m,s) ,n,1,2)$

"Ricci tensor RIC";
Ric(m,q) := sum( RII(s,m,s,q), s,1,2);

RIC : matrix( [Ric(1,1),Ric(1,2)], [Ric(2,1),Ric(2,2)] )$

"Ricci scalar curvature R";
R : sum( sum( gu[i,j]*Ric(i,j) ,j,1,2) ,i,1,2);

define( Kintrinsic(u,v),
       block([Ch111: ChrII(1,1,1), Ch121: ChrII(1,2,1), Ch211: ChrII(2,1,1),
              Ch221: ChrII(2,2,1), Ch222: ChrII(2,2,2)],
              1/g[1,1]* (diff(Ch211,v) - diff(Ch221,u) +
              Ch221*(Ch111-Ch221) + Ch211*(Ch222-Ch121))));

"Covariant derivative of vector fields";
covD(W,V) := [ V[1]*diff(W[1],u) + V[2]*diff(W[1],v) +
                sum( sum( V[i]*W[j]*ChrII(1,i,j), j,1,2) , i,1,2) ,
                V[1]*diff(W[2],u)+V[2]*diff(W[2],v) +
                sum( sum( V[i]*W[j]*ChrII(2,i,j), j,1,2) , i,1,2) ];

```

## 13 Bibliography

### References

- [1] BÄR, C. (2013): *Differential Geometry*.  
url: [https://www.math.uni-potsdam.de/fileadmin/user\\_upload/Prof-Geometrie/Dokumente/Lehre/Lehrmaterialien/skript-DiffGeo-engl.pdf](https://www.math.uni-potsdam.de/fileadmin/user_upload/Prof-Geometrie/Dokumente/Lehre/Lehrmaterialien/skript-DiffGeo-engl.pdf)
- [2] BANCHOFF, T. F. & LOVETT, S. (32022): *Differential Geometry of Curves and Surfaces*. London: Chapman and Hall.
- [3] CARMESIN, H.-O. (2012): *Schüler entdecken die Einstein-Geometrie mit dem Beschleunigungssensor*. url: [http://hans-otto.carmesin.org/images/Physikdid/2012Einstein geometrie\\_In\\_PhysicsDid\\_B.pdf](http://hans-otto.carmesin.org/images/Physikdid/2012Einstein geometrie_In_PhysicsDid_B.pdf)
- [4] CASEY, J. (1996): *Exploring Curvature*. Braunschweig: Vieweg.
- [5] COXETER, H. S. M. (1963): *Unvergängliche Geometrie*. Basel: Birkhäuser.
- [6] DESERNO, M. (2004): *Notes on Differential Geometry*.  
url: [https://www.cmu.edu/biolphys/deserno/pdf/diff\\_geom.pdf](https://www.cmu.edu/biolphys/deserno/pdf/diff_geom.pdf)
- [7] DINEEN, S. (2001): *Multivariate Calculus and Geometry*. London: Springer.
- [8] EARL, M. (2004): *The Local Theory of Curves and Surfaces*.  
url: [https://courses-archive.maths.ox.ac.uk/node/view\\_material/48844](https://courses-archive.maths.ox.ac.uk/node/view_material/48844)
- [9] GALLOWAY, ?. (2022): *Introduction to Differential Geometry*.  
url: <https://www.math.miami.edu/~galloway/IntroDGnotes.pdf>
- [10] GLUCK, H. (2022): *Intrinsic Geometry of Surfaces*.  
url: <https://www.math.upenn.edu/~shiydong/Math501X-4-IntrinsicGeometry.pdf>
- [11] GRINFELD, P. (2013): *Introduction to Tensor Analysis and the Calculus of Moving Surfaces*. New York: Springer.
- [12] HERTRICH-JEROMIN, U. (2019): *Differential Geometry*.  
url: <https://www.geometrie.tuwien.ac.at/hertrich-jeromin/pub/dg.pdf>
- [13] HECKMANN, G. (2019): *Classical Differential Geometry*.  
url: <https://www.math.ru.nl/~heckman/CDG.pdf>
- [14] HITCHIN, N. (2019): *Geometry of Surfaces*.  
url: <https://people.maths.ox.ac.uk/~joyce/Nairobi2019/Hitchin-GeometryOfSurfaces.pdf>
- [15] KREYSZIG, E. (1957): *Differentialgeometrie*. Leipzig: Akademische Verlagsgesellschaft Geest & Portig.
- [16] LINDNER, W. (2021): *Some Tapas of Tensors*.  
url: <https://georgeweight.github.io/Lindner/Some-Tapas-of-Tensors.pdf>

- [17] LINDNER, W. (2023): *Elementary Differential Geometry using EigenMath*.  
url: <https://georgeweigt.github.io/Lindner/Differential-Geometry-of-Surfaces.pdf>
- [18] LINDNER, W. (2023): *Elementary Differential Geometry using EigenMath – solutions manual*. url: <https://georgeweigt.github.io/Lindner/Differential-Geometry-of-Surfaces-solutions.pdf>
- [19] LINDNER, W. (2021): *Selected Solutions to Bachman's Differential Forms*.  
url: <https://georgeweigt.github.io/Lindner/Bachman-Differential-Forms-Solutions-selected.pdf>
- [20] LIPSCHUTZ, M. M. (1969): *Differential Geometry*. New York: McGraw-Hill
- [21] MALKOWSKY, E. & NICKEL, W. (1993): *Computergrafik in der Differentialgeometrie*. Braunschweig: Vieweg.
- [22] MARSDEN, J. & WEINSTEIN, A. (2<sup>nd</sup>1985): *Calculus III*. New York: Springer.
- [23] OLOFF, R. (6<sup>th</sup>2018): *Geometrie der Raumzeit*. [engl.: Geometry of Spacetime.] Berlin: Springer Spectrum.
- [24] PRESSLEY, A. (2001): *Elementary Differential Geometry*. London: Springer.
- [25] RECKZIEGEL, H. ET AL. (1998): *Elementare Differentialgeometrie mit Maple*. Braunschweig: Vieweg.
- [26] REJBRAND, A. (2023): *The Rejbrand Encyclopaedia of Curves and Surfaces*.  
url: <https://treks.se/surfaces.php>
- [27] SEEBURGER, P. (2018): *CalcPlot3D app*.  
url: <https://c3d.libretexts.org/CalcPlot3D/index.html>
- [28] SEEBURGER, P. (2018): *CalcPlot3D Manual*.  
url: <https://c3d.libretexts.org/CalcPlot3D/CalcPlot3D-Help/front.html>
- [29] SEEBURGER, P. (2018): *CalcPlot3D - Introduction Video*.  
url: <https://c3d.libretexts.org/CalcPlot3D/CalcPlot3D-Help/section-1.html>
- [30] SHIFRIN, T. (2018): *Differential Geometry: A First Course in Curves and Surfaces*.  
url: <https://math.franklin.uga.edu/sites/default/files/inline-files/ShifrinDiffGeo.pdf>
- [31] SNYGG, J. (2001): *A New Approach to Differential Geometry using Clifford's Geometric Algebra*. New York: Springer-Birkhäuser.
- [32] SOCHI, T. (2017): *Introduction to Differential Geometry of Space Curves and Surfaces*. London: CreativeSpace. ISBN 9781546735892.
- [33] SOFI, G. M. (2018): *Fundamental Theorem of Surfaces in  $\mathbb{R}^3$* .  
url: [https://www.cukashmir.ac.in/cukashmir/User\\_Files/image/module203\\_MMT\\_C204.pdf](https://www.cukashmir.ac.in/cukashmir/User_Files/image/module203_MMT_C204.pdf)

- [34] SPIEGEL, M. R. (1963): *Advanced Calculus*.  
New York: McGraw-Hill (Schaums' Outline Series)
- [35] STEEB, W.-H & LEWIEN, D. (1991): *Algorithms and Computation with REDUCE*.  
Mannheim: Bibliographisches Institut.
- [36] STEEB, W.-H & SHI, T. K. (1998): *Symbolic C++: An Introduction to Computer Algebra Using Object-Oriented Programming*. Singapore: Springer.
- [37] THORPE, J. A. (1979): *Elementary Topics in Differential Geometry*.  
New York: Springer.
- [38] TOTH, V. T. (2004): *Maxima - ctensor package funstions*.  
url: [https://maxima.sourceforge.io/docs/manual/maxima\\_128.html](https://maxima.sourceforge.io/docs/manual/maxima_128.html)
- [39] TOTH, V. T. (2004): *Tensor manipulation in GPL Maxima*.  
url: <https://arxiv.org/abs/cs/0503073>
- [40] VAS, L. (2021): *Surfaces I*.  
url: <https://liavas.net/courses/math430/files/Surfaces.pdf>
- [41] VAS, L. (2021): *Surfaces II*.  
url: [https://liavas.net/courses/math430/files/Surfaces\\_part2.pdf](https://liavas.net/courses/math430/files/Surfaces_part2.pdf)
- [42] VAS, L. (2021): *Surfaces III*.  
url: [https://liavas.net/courses/math430/files/Surfaces\\_part3.pdf](https://liavas.net/courses/math430/files/Surfaces_part3.pdf)
- [43] WEIGT, G. (2021): *EIGENMATH Homepage*.  
url: <https://georgeweigt.github.io>
- [44] WEIGT, G. (2021): *EIGENMATH online*.  
url: <https://georgeweigt.github.io/eigenmath-demo.html>
- [45] WEIGT, G. (2021): *The Schwarzschild Metric*.  
url: <https://georgeweigt.github.io/schwarzschild-metric.html>
- [46] WHEELER, N. (2015): *Differential Geometry of Some SURFACES IN 3-SPACE*.  
url: <https://www.reed.edu/physics/faculty/wheeler/documents/Miscellaneous%20Math/Differential%20Geometry/A.%20Surfaces%20in%203-Space.pdf>
- [47] WOLFRAM mathworld: Catenoid.  
url: <https://mathworld.wolfram.com/Catenoid.html>
- [48] WOLFRAM mathworld: Cone.  
url: <https://mathworld.wolfram.com/Cone.html>
- [49] WOLFRAM mathworld: Cylinder.  
url: <https://mathworld.wolfram.com/Cylinder.html>

- [50] WOLFRAM mathworld: ENNEPER's Surface.  
url: <https://mathworld.wolfram.com/EnnepersMinimalSurface.html>
- [51] WOLFRAM mathworld: Helicoid.  
url: <https://mathworld.wolfram.com/Helicoid.html>
- [52] WOLFRAM mathworld: MOEBIUS Strip.  
url: <https://mathworld.wolfram.com/Catenoid.html>
- [53] WOLFRAM mathworld: PLÜCKER's conoid.  
url: <https://mathworld.wolfram.com/PlueckersConoid.html>
- [54] WOLFRAM mathworld: SCHERK's Surface.  
url: <https://mathworld.wolfram.com/ScherksMinimalSurfaces.html>
- [55] WOLFRAM mathworld: Sphere.  
url: <https://mathworld.wolfram.com/Sphere.html>