Differential Geometry - Exercise 1

Florian Bogner

May 8, 2020

1. *Let X and Y be tangent vector fields on a smooth surface $M \in \mathbb{R}^3$. Show that the vector field $D_X Y - D_Y X$ along M is also a tangent vector field.*

   Let N be a unit normal field. Since X and Y are tangent fields, we have for every point p that $\langle X, N \rangle = \langle Y, N \rangle = 0$. Thus also the directional derivative of these terms is 0. By the Leibniz rule we have:

   $$0 = D_X \langle Y, N \rangle = \langle D_X Y, N \rangle + \langle Y, D_X N \rangle$$
   $$0 = D_Y \langle X, N \rangle = \langle D_Y X, N \rangle + \langle X, D_Y N \rangle$$

   Subtracting the two equation from each other and rearranging gives us:

   $$\langle D_X Y - D_Y X, Z \rangle = \langle X, D_Y N \rangle - \langle Y, D_X N \rangle$$

   We want the left side to be 0, as then the vector field in question would be orthogonal to the unit normal, thus proving it to be tangent. Luckily Theorem 3.11. tells us the right side is 0. □

2. *Consider the surface*

   $$\sigma(u, v) = (u - \frac{u^3}{3} + uv^2, v - \frac{v^3}{3} + u^2 v, u^2 - v^2)$$

   (a) *Show that the surface is conformally parametrized.*

   (b) *Show that it has zero mean curvature (do only as much computation as needed.*

   For (a) Theorem 2.45. tells us to look at the first fundamental form and see if it is a multiple of the identity matrix. For (b) we use Lemma 3.18. Again we use *sympy* ~~because we are lazy~~ to calculate the first and second fundamental form. The source code is on the last page.

   $$E = \left(u^2 + v^2 + 1\right)^2$$
   $$F = 0$$
   $$G = \left(u^2 + v^2 + 1\right)^2$$
   $$L = 2$$
   $$M = 0$$
   $$N = -2$$

   Indeed the first fundamental form is a multiple of the identity matrix since $E = G$ and $F = 0$. Thus σ is conformally parametrized. □

The mean curvature H is

$$
\begin{aligned}
H &= \frac{\det\left(\begin{pmatrix} E & F \\ F & G \end{pmatrix}, \begin{pmatrix} L & M \\ M & N \end{pmatrix}\right)}{\det\begin{pmatrix} E & F \\ F & G \end{pmatrix}} \\
&= \frac{\frac{1}{2}\left(EN + GL - FM - FM\right)}{EG - F^2} \\
&= \frac{\frac{1}{2}\left(-2E + 2E - 0 - 0\right)}{EG - F^2} \\
&= 0
\end{aligned}
$$

$\square$

3. *A vector is said to be* asymtotic *if* $II(X, X) = 0$.

   (a) *Show that if the surface M contains the line $p + tX$, then the vector X is asymptotic.*
   By definition and by Leibnitz rule we have:

   $$
   \begin{aligned}
   II(X, X) &= -\left\langle D_X N, X\right\rangle \\
   &= -D_X \left\langle N, X\right\rangle + \left\langle N, D_X X\right\rangle \\
   &= -D_X 0 + \left\langle N, 0\right\rangle \\
   &= 0
   \end{aligned}
   $$

   For the directional derivatives let $\gamma(t) = p + tX$. The first summand is 0 because the unit normal vector field N is, well, normal to vectors inside the surface such as X. The second summand is 0 as $D_X X$ is zero because X is constant with respect to t. The direction of a line is always the same. $\square$

   (b) *Assume that the second fundamental form is non-degenerate and indefinite at p, so that it has two different asymptotic directions. Show that these directions are orthogonal if and only if the mean curvature vanishes at p.*
   Let X and Y be to two asymptotic vectors representing the two different directions. Since they are asymptotic, i.e. $\langle S(X), X\rangle = 0$, we have that $(X, S(X))$ is a (orthogonal) basis. Lets write the shape operator in this basis:

   $$
   S = \begin{pmatrix} 0 & a \\ 1 & b \end{pmatrix}
   $$

   The characteristic polynomial is $t^2 - bt - a = 0$. Its roots are the principal curvatures. The mean curvature is zero iff the roots are symmetric around 0 which is the case iff $b = 0$ which is equivalent to $S(Y) = aX$. Since $S(Y) \perp Y$, this is the case iff $X \perp Y$. $\square$

4. *Show that the tangent developable of every curve has Gauss curvature zero.*

Let γ be a curve. Its tangent developable is $\sigma(u, v) = \gamma(u) + v \cdot \dot{\gamma}(u)$. For brevitiy we will omit arguments of u. To use Lemma 3.18. we calculate (parts of) the second fundamental form:

$$
\begin{aligned}
\sigma_u &= \dot{\gamma} + v\ddot{\gamma} \\
\sigma_v &= \dot{\gamma} \\
\sigma_{uu} &= \text{unimportant} \\
\sigma_{uv} &= \ddot{\gamma} \\
\sigma_{vv} &= 0 \\
\mathcal{N} = \frac{\sigma_u \times \sigma_v}{\|\sigma_u \times \sigma_v\|} &= c \cdot (\dot{\gamma} \times \ddot{\gamma}) \quad \text{for some } c \\
L = \langle \sigma_{uu}, \mathcal{N} \rangle &= \text{unimportant} \\
M = \langle \sigma_{uu}, \mathcal{N} \rangle &= c \cdot \langle \ddot{\gamma}, \dot{\gamma} \times \ddot{\gamma} \rangle = 0 \\
N = \langle \sigma_{uu}, \mathcal{N} \rangle &= 0
\end{aligned}
$$

Thus the Gauss curvature is

$$
K = \frac{\det \begin{pmatrix} L & 0 \\ 0 & 0 \end{pmatrix}}{\det I} = 0
$$

$\square$

```
import sympy as sp
from sympy import diff, sqrt

class Vec3:
    def __init__(self, x, y, z):
        self.x = x
        self.y = y
        self.z = z

    def __add__(self, other):
        return Vec3(self.x + other.x,
                    self.y + other.y,
                    self.z + other.z)

    def __mul__(self, scalar):
        return Vec3(scalar * self.x,
                    scalar * self.y,
                    scalar * self.z)

    __rmul__ = __mul__

    def __truediv__(self, scalar):
        return Vec3(self.x / scalar,
                    self.y / scalar,
                    self.z / scalar)

    def diff(self, var):
        return Vec3(diff(self.x, var),
                    diff(self.y, var),
                    diff(self.z, var))

    def dot(self, other):
        return self.x * other.x + self.y * other.y + self.z * other.z

    def cross(self, other):
        return Vec3(self.y * other.z - self.z * other.y,
                    self.z * other.x - self.x * other.z,
                    self.x * other.y - self.y * other.x)

    def norm(self):
        return sqrt(self.x**2 + self.y**2 + self.z**2)

sp.init_printing(forecolor = 'White')
u, v = sp.symbols("u v", real = True)

sig = Vec3(u - u**3/3 + u * v**2,
           v - v**3/3 + u**2 * v,
           u**2 - v**2)
```

```
sig_u = sig.diff(u)
sig_v = sig.diff(v)

sig_uu = sig.diff(u).diff(u)
sig_uv = sig.diff(u).diff(v)
sig_vv = sig.diff(v).diff(v)

c = sig_u.cross(sig_v)

NN = c / c.norm()

E = sig_u.dot(sig_u)
F = sig_u.dot(sig_v)
G = sig_v.dot(sig_v)

L = sig_uu.dot(NN)
M = sig_uv.dot(NN)
N = sig_vv.dot(NN)

print("\\begin{align*}")
for l in "EFGLMN":
    print(l,"&=",sp.latex(sp.simplify(sp.factor(sp.expand(sp.simplify(eval(l)))))), "\\\\")
print("\\end{align*}")
```