Math 206, Spring 2007, Computer Lab 3

1. Introduction

This lab, like the others, is simply supposed to demonstrate how Mathematica can help you with differential geometry calculations. The focus here is on writing functions — in other words, short programs. If you find this difficult, you should carefully review the examples here and in the earlier labs, to make sure you understand how they work. Or talk to me about it.

You are welcome to use these and other Mathematica functions on other assignments. Some are relevant to this week's homework, for example. If you do use Mathematica for homework, then do not simply quote the answer given by Mathematica; include all of the code needed to reproduce the answer, so that the grader understands and believes your solution.

Always hand in labs separately from homework.

2. Some Functions for Surfaces

Here are some functions I wrote for surfaces. I've been using these (and the ones you're going to write) to generate examples for class.

```
normSquared[v_] := Dot[v, v];
norm[v_] := Sqrt[normSquared[v]];
x1[x_] := D[x, u1];
x2[x_] := D[x, u2];
x11[x_] := D[D[x, u1], u1];
x12[x_] := D[D[x, u1], u2];
x21[x_] := D[D[x, u2], u1];
x22[x_] := D[D[x, u2], u2];
rawNormal[x_] := Cross[x1[x], x2[x]];
normal[x_] := rawNormal[x] / norm[rawNormal[x]];
eLarge[x_] := Dot[x1[x], x1[x]];
fLarge[x_] := Dot[x1[x], x2[x]];
gLarge[x_] := Dot[x2[x], x2[x]];
eSmall[x_] := -Dot[D[normal[x], u1], x1[x]];
fSmall[x_] := -Dot[D[normal[x], u2], x1[x]];
gSmall[x_] := -Dot[D[normal[x], u2],  x2[x]];
gaussCurv[x_] := (eSmall[x] gSmall[x] - fSmall[x] fSmall[x])
               / (eLarge[x] gLarge[x] - fLarge[x] fLarge[x]);
```

They should be pretty self-explanatory. As an example, here is how you use the `gaussCurv` function to compute the Gaussian curvature of the sphere as parametrized by stereographic projection:

```
stereo[u1_, u2_] := {(2u1) / (1 + u1^2 + u2^2),
                     (2u2) / (1 + u1^2 + u2^2),
                     (u1^2 + u2^2 - 1) / (1 + u1^2 + u2^2)};
Simplify[gaussCurv[stereo[u1, u2]]]
```

3. A Linear Algebra Tip

Suppose we have a 2×2 matrix

$$m = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

and a 2-dimensional vector z and we want to solve $my = z$. Assuming that m is invertible, its inverse is

$$m^{-1} = \frac{\begin{bmatrix} d & -b \\ -c & a \end{bmatrix}}{ad - bc}.$$

So $y = m^{-1}z$. But how do we enter all of this into Mathematica? First, do

```
mInv = {{d, -b}, {-c, a}} / (a d - b c);
```

We want z to be a 2×1 column matrix, so we should enter it not as `{z1, z2}` (which isn't a matrix at all) but as

```
z = {{z1}, {z2}}
```

Now after we do `y = mInv z`, `y` is the 2×1 column matrix

```
{{(d z1 - bz2) / (a d - b c)}, {(-c z1 + a z2) / (a d - b c)}}
```

You can pick off the first entry in `y` as `y[[1, 1]]` and the second entry as `y[[2, 1]]`. Alternatively, if you want these two entries in a simple list, like

```
{(d z1 - bz2) / (a d - b c), (-c z1 + a z2) / (a d - b c)}
```

you can get that list by taking

```
(Transpose[y])[[1]]
```

This turns the column matrix `y` into a row matrix, and then picks off the first row, as a vector.

4. Assignment

As in the previous lab, just hand in what's asked for here, in a polished form. You should test your functions on a few examples such as parametrizations of the sphere, torus, cylinder, etc., but do not hand in these tests. Just hand in the definitions of the functions, and explanatory text if you think it's needed.

4.1. **Mean Curvature.** Write a `meanCurv` function, similar to the `gaussCurv` function above, to compute the mean curvature.

4.2. **Principal Curvatures.** Remember that braces `{, }` delimit lists in Mathematica. You can use them to write a function that returns not just one item, but a whole list of items. In this problem, write a function to return both principal curvatures k_1 and k_2. It should be of the form

```
princCurv[x_] := {..., ...};
```

where the first ... is k_1 and the second is k_2. Hint: Use the `gaussCurv` and `meanCurv` functions in there.

4.3. **Christoffel Symbols.** Write a function

```
chris[x_] := {{{..., ...}, {..., ...}}, {{..., ...}, {..., ...}}};
```

where the ... represent the Christoffel symbols Γ^1_{11}, Γ^2_{11}, Γ^1_{12}, Γ^2_{12}, Γ^1_{21}, Γ^2_{21}, Γ^1_{22}, Γ^2_{22}. (By putting them in this order, you can then do stuff like

```
ch = chris[stereo[u1, u2]];
ch[[1]][[2]][[1]]
ch[[1]][[2]][[2]]
```

to get Γ^1_{12} and Γ^2_{12}, for example. You can also use the slightly shorter form `ch[[1, 2, 1]]` instead of `ch[[1]][[2]][[1]]`, to save typing so many brackets.)

4.4. **Gauss Formula.** Write a function `gaussCurvNew` that computes the Gaussian curvature using the Gauss formula explained on pages 234-235 instead of the $(eg - f^2)/(EG - F^2)$ formula.

4.5. **Time Spent.** Please tell me how many hours you spent on this lab altogether. The answer does not affect your grade.