Curves and Surfaces in *Mathematica*
Lecture 10/19/2005

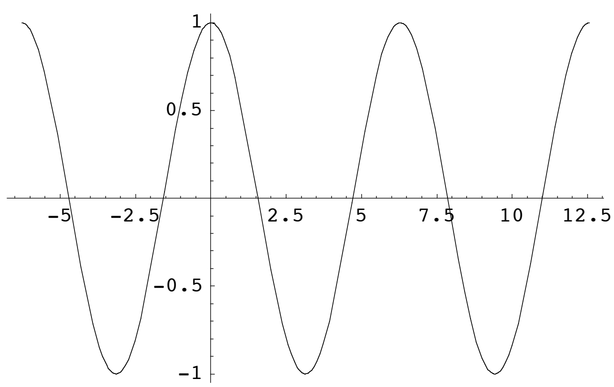
In[1]:= **1 + 1**

Out[1]= 2

Plotting Curves

▪ Plotting the graph of Cos[x]

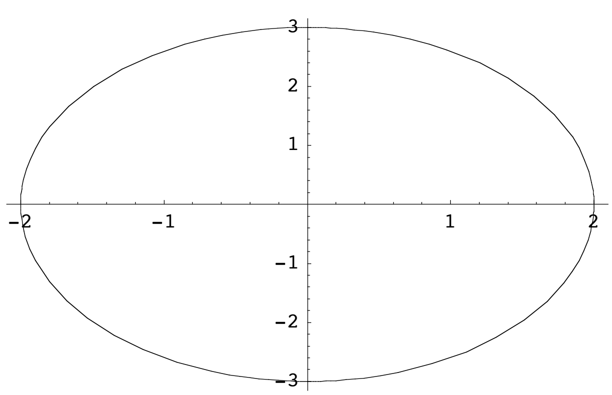
In[2]:= **Plot[Cos[x], {x, -2 Pi, 4 Pi}]**



Out[2]= -Graphics-

▪ Plotting a parametric representation of an ellipse

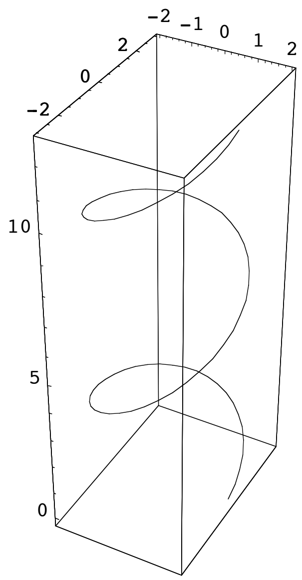
In[4]:= **ParametricPlot[{2 Cos[t], 3 Sin[t]}, {t, 0, 2 Pi}]**



Out[4]= -Graphics-

▪ Plotting a parametric representation of a helix

```
In[5]:= ParametricPlot3D[{2 Cos[t], 3 Sin[t], t}, {t, 0, 4 Pi}]
```




```
Out[5]= -Graphics3D-
```

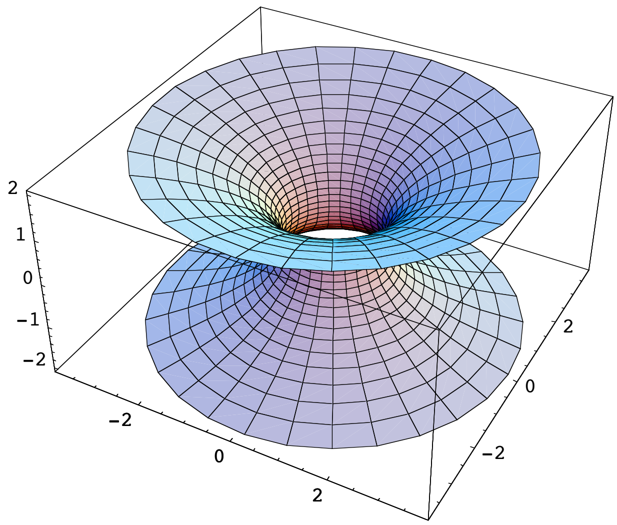
The Helicoid

▪ Here's a surface patch for the Helicoid

```
In[6]:= x[u_, v_] := {Cos[u] Cosh[v], Sin[u] Cosh[v], v}
```

▪ Let's plot it

```
In[7]:= ParametricPlot3D[x[u, v], {u, 0, 2 Pi}, {v, -2, 2}]
```




```
Out[7]= -Graphics3D-
```

▪ Is x[u,v] a regular surface patch?

```
In[8]:= D[x[u, v], u]

Out[8]= {-(Cosh[v] Sin[u]), Cos[u] Cosh[v], 0}

In[9]:= D[x[u, v], v]

Out[9]= {Cos[u] Sinh[v], Sin[u] Sinh[v], 1}

In[10]:= Cross[D[x[u, v], u], D[x[u, v], v]]

Out[10]= {Cos[u] Cosh[v], Cosh[v] Sin[u],
            -(Cos[u]^2 Cosh[v] Sinh[v]) - Cosh[v] Sin[u]^2 Sinh[v]}

In[11]:= Simplify[%]

Out[11]= {Cos[u] Cosh[v], Cosh[v] Sin[u], -(Cosh[v] Sinh[v])}

In[12]:= % . %

Out[12]= Cos[u]^2 Cosh[v]^2 + Cosh[v]^2 Sin[u]^2 + Cosh[v]^2 Sinh[v]^2

In[13]:= Simplify[%]

Out[13]= Cosh[v]^4
```

■ **The last expression never vanishes, hence, x[u,v] is a surface patch.**

■ **Let's calculate the cross product using previous output to demonstrate the syntax.**

```
In[16]:= Simplify[Cross[%8, %9].Cross[%8, %9]]

Out[16]= Cosh[v]^4
```

■ **Here's how to get rid of unwanted square roots**

```
In[17]:= Sqrt[a^2]

Out[17]= Sqrt[a^2]

In[18]:= Simplify[%]

Out[18]= Sqrt[a^2]

In[19]:= PowerExpand[%]

Out[19]= a
```

■ **Back to the Helicoid. Let's calculate the "E" component of the first fundamental form.**

```
In[20]:= x[u, v]

Out[20]= {Cos[u] Cosh[v], Cosh[v] Sin[u], v}

In[21]:= D[x[u, v], u] . D[x[u, v], u]

Out[21]= Cos[u]^2 Cosh[v]^2 + Cosh[v]^2 Sin[u]^2

In[22]:= Simplify[%]

Out[22]= Cosh[v]^2
```

■ **Thus, the above is E. The expression for F is**

```
In[23]:= Simplify[D[x[u, v], u] . D[x[u, v], v]]

Out[23]= 0
```

■ The expression for G is

```
In[24]:= Simplify[D[x[u, v], v] . D[x[u, v], v]]
```

```
Out[24]= Cosh[v]^2
```

Let ' s define a function which automatically calculates E. We will call the function EE since E is a reserved word which means the number 2.718281828 ...

```
In[28]:= EE[x_, u_, v_] := Simplify[ D[x[u, v] , u] . D[x[u, v], u]]
```

```
In[29]:= x[u, v]
```

```
Out[29]= {Cos[u] Cosh[v], Cosh[v] Sin[u], v}
```

■ Applying EE to the case of the Helicoid, we obtain

```
In[30]:= EE[x, u, v]
```

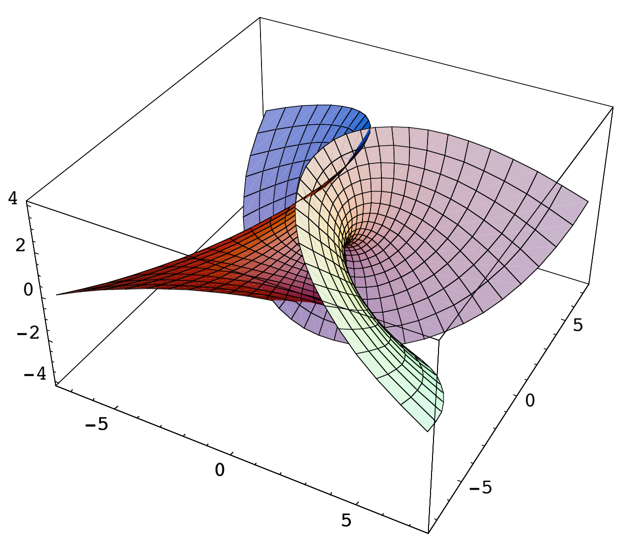
```
Out[30]= Cosh[v]^2
```

Enneper's Surface

```
In[31]:= x[u_, v_] := {u - u^3 / 3 + u v^2, v - v^3 / 3 + v u^2, u^2 - v^2}
```

■ Notice that the following fails to be 1-1 because of the self-intersection

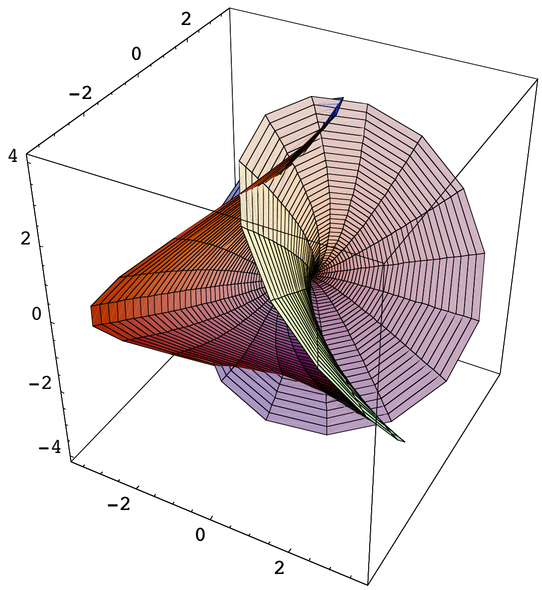
```
In[35]:= ParametricPlot3D[x[u, v], {u, -2, 2}, {v, -2, 2}]
```




```
Out[35]= -Graphics3D-
```

- **Let's plot Enneper's surface but parametrized differently. We insert the PlotPoints → {50,40} argument in order to smooth out the plot.**

```
In[41]:= ParametricPlot3D[x[ r Cos[t], r Sin[t]], {r, 0, 2}, {t, 0, 2 Pi}, PlotPoints → {50, 40}]
```




```
Out[41]= -Graphics3D-
```