

# Team Exercise #9 - Vector Fields II

John A. Pelesko

May 3, 2004

The automatic mechanical manipulation of micro- and macro-scale parts presents a complicated engineering challenge. One solution that has been explored for the manipulation of micro-scale parts involves the use of a large array of individual actuators. To visualize this system, think of an air hockey table. At each point on the table, where the air is typically forced straight upwards, imagine a very small nozzle that can direct the flow and hence the force in any direction. If one wanted to plan the manipulation of parts on a 2-d region using this system, one can visualize formulating a vector field over some region of space. The vector field specifies the magnitude and direction of the “push” at each point. We ignore the push in the third dimension and simply imagine that the parts are supported on a cushion of air (like the air hockey table). If a particle (or air hockey puck) is placed on the table we would like to know how it moves. A reasonable model would say that its path, given by a position vector,  $\vec{r}(t) = \langle x(t), y(t) \rangle$ , is such that at each point the tangent to the path is given by the vector field. To get a handle on this, consider the vector field:

$$\vec{F} = \langle -y, x \rangle .$$

## (1) Plot this vector field.

The tangent vector to the path is  $\frac{d\vec{r}}{dt}$ . By our assumption above to obtain the motion, we set

$$\frac{d\vec{r}}{dt} = \vec{F} .$$

This is a system of *ordinary differential equation*. Maple may be used to find the path. First, we tell Maple what our equations are and where to start the path:

```
> dsys1 := {diff(x(t),t)=-y(t),diff(y(t),t)=x(t),x(0)=1,y(0)=0};  
          dsys1 := { $\frac{d}{dt}x(t) = -y(t)$ ,  $\frac{d}{dt}y(t) = x(t)$ ,  $x(0) = 1$ ,  $y(0) = 0$ }
```

Next, we tell Maple to solve for the path:

```
> dsol1 :=  
> dsolve(dsys1,{x(t),y(t)},type=numeric,output=listprocedure);  
  
dsol1 := [t = (proc(t) ... end proc), x(t) = (proc(t) ... end proc),  
          y(t) = (proc(t) ... end proc)]
```

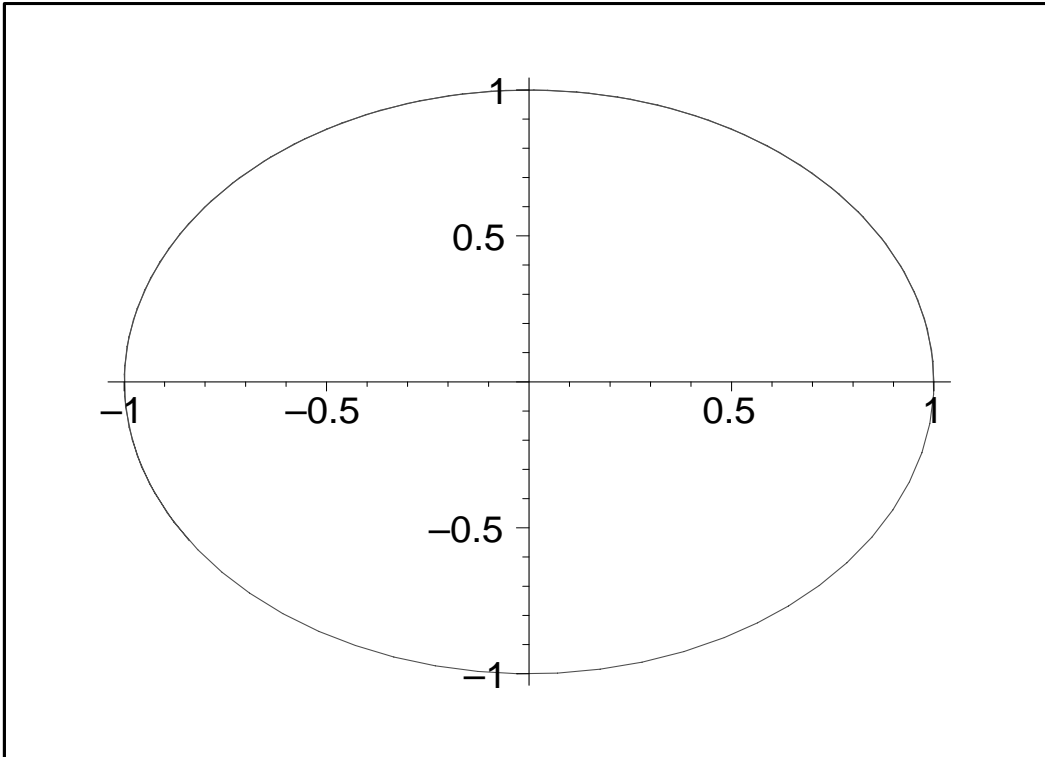
Now, we evaluate the output functions, in this case the path coordinates, so that we may plot:

```
> X := eval(x(t), %);  
          X := proc(t) ... end proc  
> Y := eval(y(t), %%);
```

$Y := \text{proc}(t) \dots \text{end proc}$

Finally, we can plot the path:

```
> plot([X,Y,0..10]);
```



**(2) Plot the vector field and the path that starts at  $x(0) = 1, y(0) = 2$  on the same plot.**

Now, as you can see, the vector field explored above is not terribly useful. It takes every part and simply rotates around the origin. Next, you need to explore some useful vector fields.

**(3) Design a vector field, consistent with our air hockey system, that moves every object to the origin. Plot the vector field and typical particle paths using Maple.**

**(4) Compute the work done by your system in moving a particle from an arbitrary starting point to the origin.**

Your vector field is more realistic, but still far from perfect. In a realistic situation you will not have a perfectly flat empty infinite plane to work with. Further, one might not only want to move objects, but also might like to orient them.

**(5) Imagine your region has a circular obstacle of radius one at the origin. Suppose you want to move individual pieces to the point  $(5, 5)$ . Design a vector field to do this. Plot the vector field and typical paths on the same plot.**