

Math 2250-4

November 12, 2001

First order systems of Differential Equations

Why you should expect existence and uniqueness for the IVP

```
> restart:with(DEtools):with(plots):with(linalg):
Warning, the name changecoords has been redefined
Warning, the name adjoint has been redefined
Warning, the protected names norm and trace have been redefined and unprotected
```

Here's the example of a first order system which we raced through in the last 200 seconds of class on Friday. It deserves a little more attention. You should think of this example as arising from the simple harmonic oscillator equation

$$\left[\left(\frac{\partial^2}{\partial t^2} y(t) \right) + y(t) = 0 \right]$$

By our conversion recipe, we let $x_1=y(t)$ and $x_2=dy/dt$. This leads to the equivalent first order system (check!)

$$\left[\begin{array}{c} \frac{dx_1}{dt} \\ \frac{dx_2}{dt} \end{array} \right] = \begin{bmatrix} x_2 \\ -x_1 \end{bmatrix}$$

Since the solutions $y(t)$ to the harmonic oscillator equation are

$$\left[y(t) = A \cos(t) + B \sin(t) \right]$$

we deduce that the solutions to the equivalent linear system are (check!)

$$\left[\begin{array}{c} x_1(t) \\ x_2(t) \end{array} \right] = A \begin{bmatrix} \cos(t) \\ -\sin(t) \end{bmatrix} + B \begin{bmatrix} \sin(t) \\ \cos(t) \end{bmatrix}$$

Suppose the initial conditions were

$$\left[\begin{array}{c} x_1(0) \\ x_2(0) \end{array} \right] = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

i.e.

$$\left[\begin{array}{c} 0 \\ 1 \end{array} \right] = A \begin{bmatrix} 1 \\ 0 \end{bmatrix} + B \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

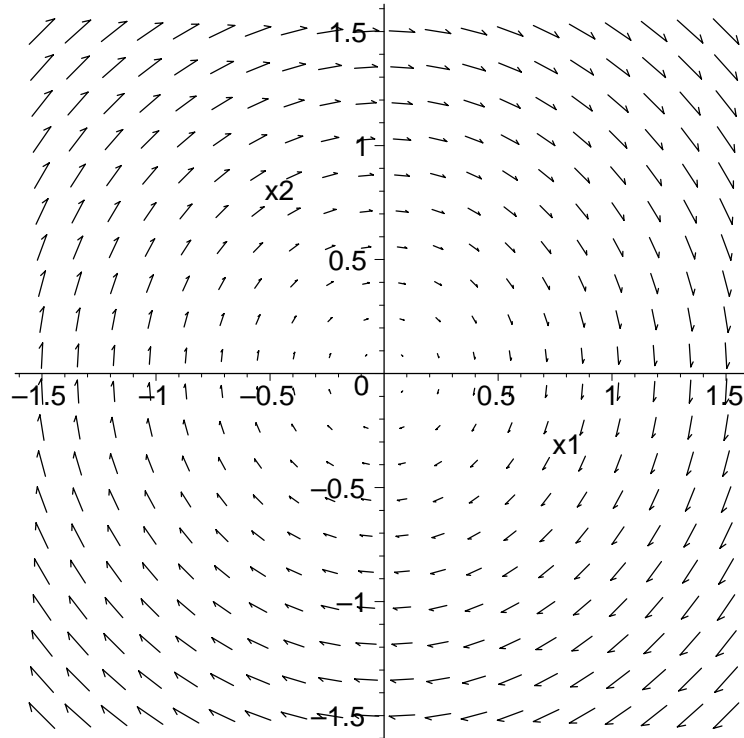
so $A=0$ and $B=1$, and our solution is

$$\left[\begin{array}{c} x_1(t) \\ x_2(t) \end{array} \right] = \begin{bmatrix} \sin(t) \\ \cos(t) \end{bmatrix}$$

Now, pretend that we didn't know the analytic solutions to a given first order system of differential equations, for example the one we did just solve analytically. We could still visualize the the velocity (or in math language, tangent) field $\mathbf{f}(\mathbf{x},t)$ from our first order system of differential equations $d\mathbf{x}/dt=\mathbf{f}(\mathbf{x},t)$, and for given initial conditions, $[x_1(t),x_2(t)]$ would be the parametric curve which started at the initial

point, and then "followed" the velocity field, in the sense that its tangent vector $d\mathbf{x}/dt$ has value exactly equal to $\mathbf{f}(\mathbf{x},t)$. The notion of tangent vector and tangent field is very important. It is very related to, but not exactly the same as, the way we visualized solution graphs to first order differential equations as following slope fields.

```
[ > with(plots):
> fieldplot([x2,-x1],x1=-1.5..1.5,x2=-1.5..1.5);
```



With the initial conditions which were specified earlier, we would start at the point $[1,0]$, and then follow the velocity field around curves which look a lot like circles. If we wanted to get a numerical approximation, we could just transpose Euler (or improved Euler or Runge-Kutta) for first order equations to first order systems: (Or we could use one of Maple's many numerical packages.)

```
[ > f1:=(x1,x2,t)->x2:
    f2:=(x1,x2,t)->-x1:
    #our right hand side F(x,t)=[f1(x,t),f2(x,t)]
    #in our first order system of DE's dx/dt = F(x,t).
> n:=100:
    #number of time steps
t0:=0: #initial time
tn:=evalf(Pi):
    #final time
h:=(tn-t0)/n:
    #time step
```

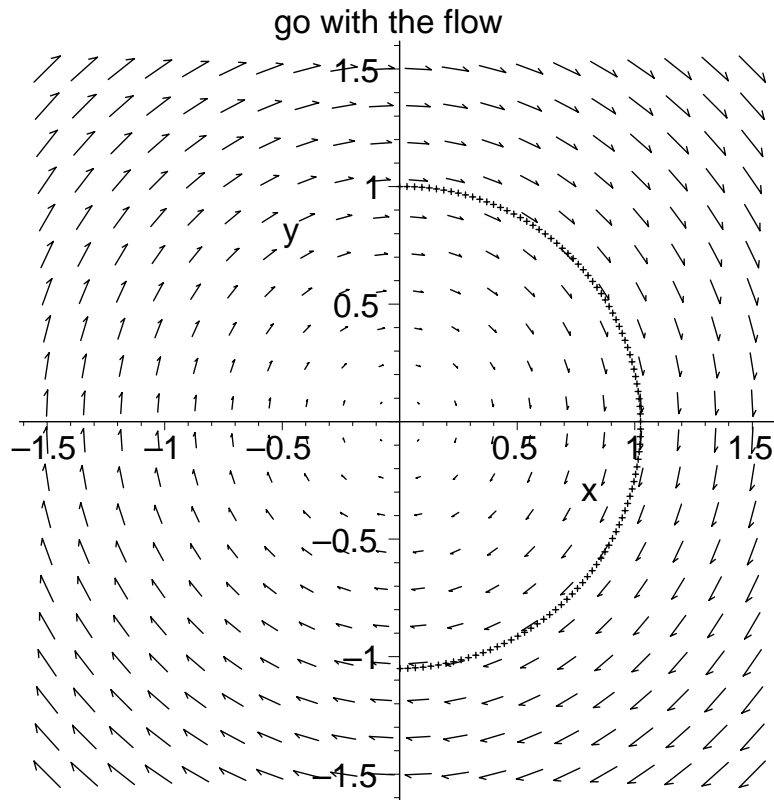
```

X1:=vector(n+1):
X2:=vector(n+1):
  #vectors to hold Euler approximates
X1[1]:=0:
X2[1]:=1:
  #initial conditions

> #Euler loop:
for i from 1 to n do
  x1:=X1[i]:
  x2:=X2[i]:
  t:=t0+(i-1)*h:
  k1:=f1(x1,x2,t):
  k2:=f2(x1,x2,t):
  #k1 and k2 are velocity components
  X1[i+1]:=x1+h*k1:
  X2[i+1]:=x2+h*k2:
  #update X1 and X2 with Euler!
od:

> veloc:=fieldplot([y,-x],x=-1.5..1.5,y=-1.5..1.5):
Eulerapprox:=pointplot({seq([X1[i],X2[i]],i=1..n+1)}):
display({veloc,Eulerapprox},title="go with the flow");

```



(1) Since any single n th order differential equation is equivalent to a related first order system of n first order differential equations, the existence and uniqueness theorem for first order systems implies the existence and uniqueness theorem that we quoted a month ago for n th order (linear) differential equations!

(2) If you want to solve an n th order differential equation numerically you first convert it to a first order system, and then use Euler or Runge-Kutta, to solve it! (Or use a package that does something like that while you aren't looking.)

(3) In physics, the conversion of a second order harmonic-oscillator type DE into the first system, and the discussion of $x_1=y$ and $x_2=dy/dt$ is called a "phase plane analysis". Notice in our picture above, the vertical axis is actually representing the velocity in the original second order deqtn. Recall that the NONLINEAR pendulum equation (for suitable $L=g$) was

```

> restart:with(DEtools):with(plots):
  with(linalg):
Warning, the name changecoords has been redefined
Warning, the name adjoint has been redefined
Warning, the protected names norm and trace have been redefined and unprotected

```

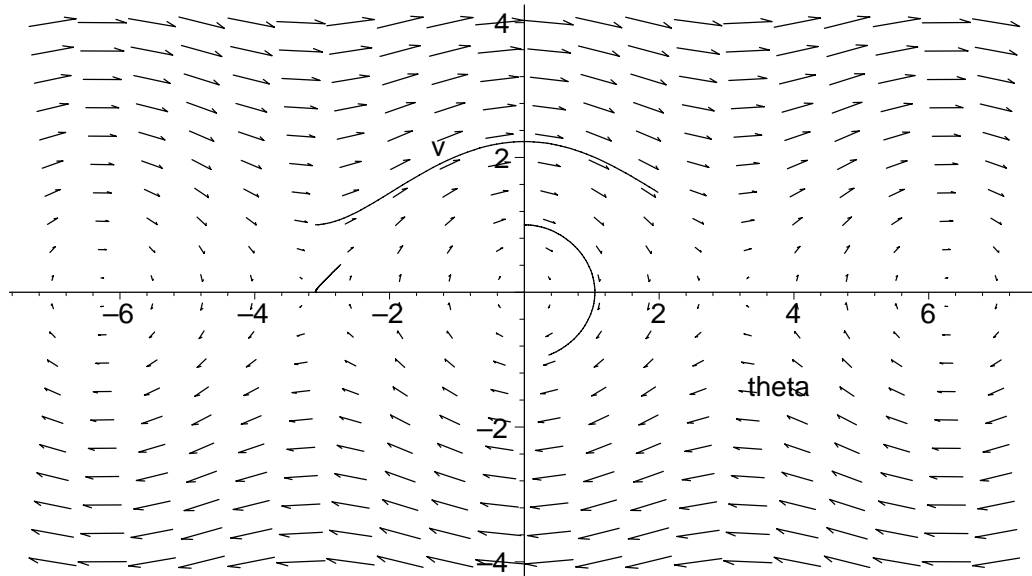
$$\left(\frac{\partial^2}{\partial t^2} y(t)\right) + \sin(y(t)) = 0$$

where $y(t)$ was actually $\theta(t)$. If we convert this to a system we get

$$\begin{bmatrix} \frac{dx1}{dt} \\ \frac{dx2}{dt} \end{bmatrix} = \begin{bmatrix} x2 \\ -\sin(x1) \end{bmatrix}$$

and this makes an interesting tangent vector field plot. Here's a picture of the velocity field, along with partial solutions to three initial value problems. Can you explain what the pendulum is doing? Also, notice there are two kinds of stationary (i.e. constant) solutions, at least according to the velocity field). Which ones would you consider "stable"? Also, notice that the picture near the origin looks like the work we did on the "linearized problem" above.

```
> velocfield:=fieldplot([v,-sin(theta)],theta=-7..7,v=-4..4):
pendeqtn1:={diff(x(t),t)=y(t),diff(y(t),t)=-sin(x(t)),
  x(0)=0,y(0)=1}:
traj1:=odeplot(dsolve(pendeqtn1,
  [x(t),y(t)], type=numeric),[x(t),y(t)],
  0..3,numpoints=25,color=black):
pendeqtn2:={diff(x(t),t)=y(t),diff(y(t),t)=-sin(x(t)),
  x(0)=-3.1,y(0)=0}:
traj2:=odeplot(dsolve(pendeqtn2,
  [x(t),y(t)], type=numeric),[x(t),y(t)],
  0..3,numpoints=25,color=black):
pendeqtn3:={diff(x(t),t)=y(t),diff(y(t),t)=-sin(x(t)),
  x(0)=-3.1,y(0)=1}:
traj3:=odeplot(dsolve(pendeqtn3,
  [x(t),y(t)], type=numeric),[x(t),y(t)],
  0..3,numpoints=25,color=black):
display({velocfield,traj1,traj2,traj3},
  title="pendulum motion");
```



[A final example:

```
> restart:
with(plots):with(linalg):
A:=matrix(2,2,[0,1,2,1]);
eigenvects(A);
fieldplot([x2,2*x1+x2],
### WARNING: incomplete quoted name; use ` to end the name
x1=-4..4,x2=-4..4,color=black,title='relate this to
eigenvectors and eigenvalues');
```

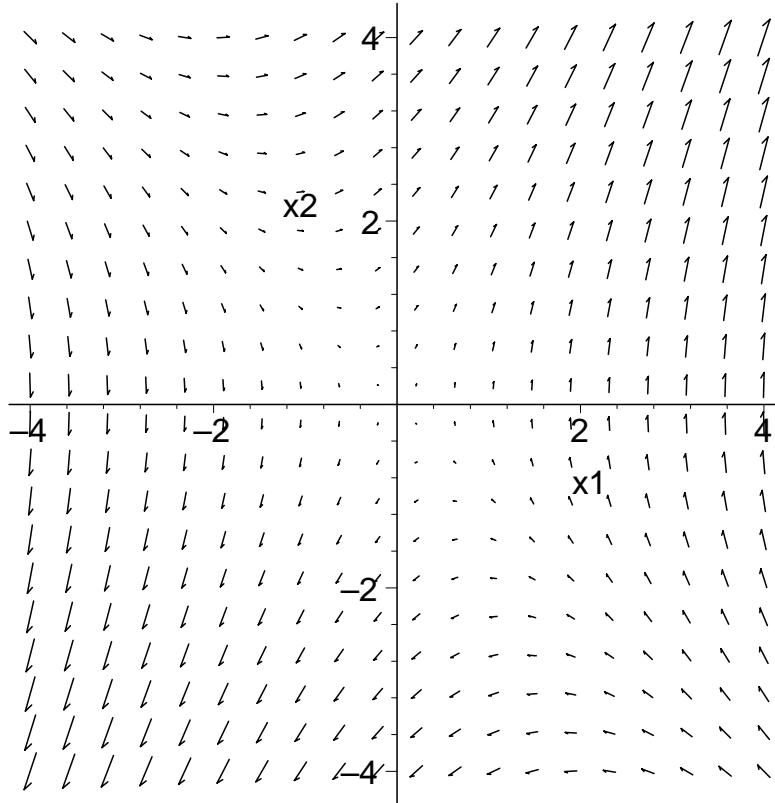
>
Warning, the name changecoords has been redefined

Warning, the protected names norm and trace have been redefined and unprotected

$$A := \begin{bmatrix} 0 & 1 \\ 2 & 1 \end{bmatrix}$$

[-1, 1, {[-1, 1]}, [2, 1, {[1, 2]}]]

relate this to
eigenvectors and eigenvalues



[>