```
>
```

flowbox.mws Calculus III or IV: geometric visulaization of divergence in 2-D

Written by Matt Miller (miller@math.sc.edu)
Modified for use in Vector Analysis (Math 550) by Doug Meade (meade@math.sc.edu)

Introduction

This Maple worksheet is designed to help the student visualize one geometric interpretation of divergence, namely as local rate of change of volume (per unit volume). We approximate the divergence by observing how a small "flowbox" behaves subject to the flow determined by the vector field.

```
> restart;
> with(plots):
> with(linalg):
> with(DEtools):
Warning, new definition for norm
Warning, new definition for trace
```

First we compute the divergence of a simple vector field by using the coordinate-dependent formula. You are going to enter other vector fields here later on.

```
> F:= ( x , y ) -> [ x - y , x + y + 0.001*x^2];
```

$$F := (x, y) \rightarrow [x - y, x + y + .001\, x^2]$$

```
> vars:=  [x, y];
```

$$vars := [x, y]$$

```
> div_F:= diverge( F(x,y), vars );
```

In particular, the divergence at the point (1.5,2.0) is:

```
> subs( x = 1.5, y = 2.0, div_F);
```

Part I: Local behavior

We specify the test box dimensions. The coordinates of the lower left hand corner are (a,b).

```
> a:= 1.5 ; b:= 2.0 ; deltax:= 0.5 ; deltay:= 0.4 ;
> aplus:=a + deltax : bplus:= b + deltay:
> corners:= [ [a, b], [aplus, b], [aplus, bplus], [a, bplus], [a, b ]
  ];
```

$$a := 1.5$$

$$b := 2.0$$

$$deltax := .5$$

$$deltay := .4$$

$$corners := [[1.5, 2.0], [2.0, 2.0], [2.0, 2.4], [1.5, 2.4], [1.5, 2.0]]$$

```
> Testbox:= polygonplot(corners, color = RED):
> Fplot:= fieldplot( F(x,y) , x = -1 .. 3 , y = -1 .. 7 ,
  arrows=SLIM):
> display({ Fplot , Testbox } );
```

To see what happens to the box in a small neighborhood of (a,b) we use the direction vectors to give the rates of change in each component separately. Notice that these rates are different at each of the four corners of the box.

```
> deltat:= 0.9;
```

deltat := .9

```
> LL:=  [a + F(a,b)[1]*deltat , b + F(a,b)[2]*deltat];
> LR:=  [aplus + F(aplus,b)[1]*deltat , b + F(aplus,b)[2]*deltat] ;
> UL:=  [a + F(a,bplus)[1]*deltat , bplus + F(a,bplus)[2]*deltat] ;
> UR:=  [aplus + F(aplus, bplus)[1]*deltat , bplus + F(aplus,
  bplus)[2]*deltat] ;
> newcorners := [ LL, LR, UR, UL, LL ];
```

LL := [1.05, 5.152025]

LR := [2.0, 5.603600]

UL := [.69, 5.912025]

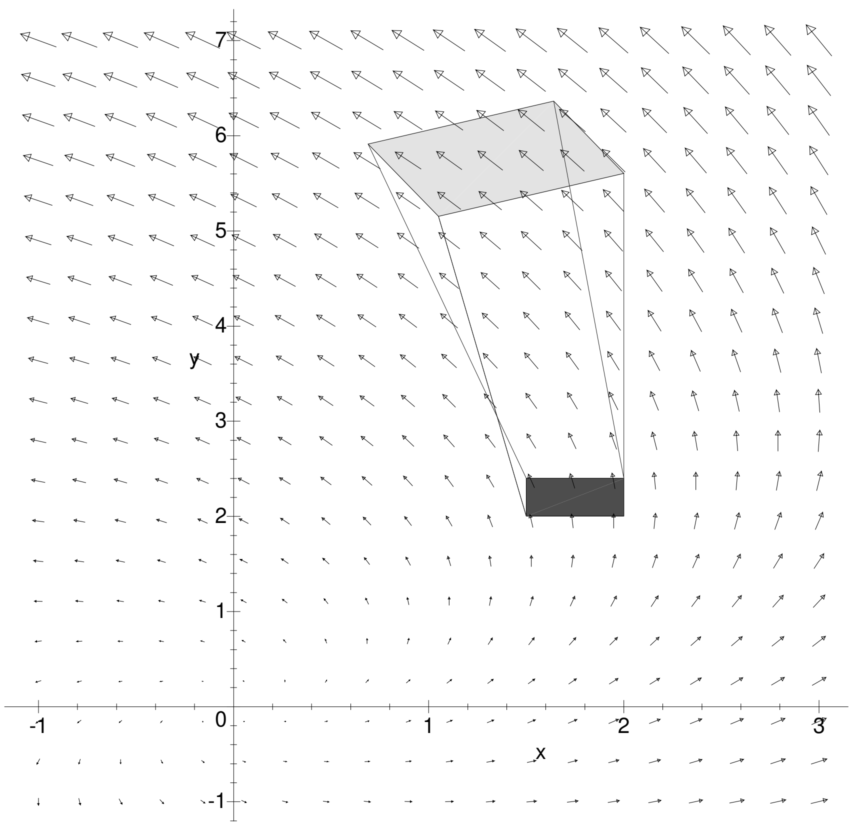
UR := [1.64, 6.363600]

newcorners :=

[[1.05, 5.152025], [2.0, 5.603600], [1.64, 6.363600], [.69, 5.912025], [1.05, 5.152025]]

```
> Newbox:= polygonplot(newcorners, color=YELLOW):
> Connect := plot( [ seq( [corners[i],newcorners[i] ],
  i=1..nops(corners) ) ], style=LINE, color=BLUE ):
> display( {Fplot, Testbox, Newbox, Connect}, title=sprintf(`One step
  deltat = %06.4g`,deltat));
```

One step deltat = 0.9000



At this point we have chosen a fairly large value for *deltat*, just so we can see the effect clearly. We will come back and repeat these calculations with smaller and smaller values of *deltat*. The area of the new parallelogram is the length of the cross product of the direction vectors for the two sides. The two sides are (in 2-d):

```
> side1:= LR - LL ;
> side2:= UL - LL ;
```

$$side1 := [.95, .451575]$$

$$side2 := [-.36, .760000]$$

Maple will compute cross products of 3-d vectors; one way to add zero as the third component of the vector is:

```
> side1a := [ op(side1), 0 ];
> side2a := [ op(side2), 0 ];
```

$$side1a := [.95, .451575, 0]$$

$$side2a := [-.36, .760000, 0]$$

The area of the new parallelogram is:

```
> A_new := norm( crossprod( side1a, side2a ), 2 );
```

$$A\_new := .8845670000$$

The area of the original parallelogram is:

```
> A_old:= deltax * deltay ;
```

$$A\_old := .20$$

Hence, the rate of expansion (per unit volume) is:

```
> rate_of_expansion:= (( A_new - A_old ) / A_old ) / deltat ;
```

$$rate\_of\_expansion := 3.803150000$$

Repeat the calculation with successively smaller values of *deltat*. How do the successive values that you obtain for the rate of expansion compare with our original computation of div F?

```
>
```

Points to Ponder

Once things seem clear try working through the same process with the same vector field, but starting with different boxes (first keep a, and b fixed, but change *deltax* and *deltay*, and again take a sequence of successively smaller values of *deltat*; then start all over again with a box located with lower left corner at a new point (a, b), and vary *deltax*, *deltay*, and *deltat*).

- Keep a record of your observations!
- What conclusions do you draw?
- What effect do the size of deltax and deltay have on the calculations?

```
>
```

Part II. Global behavior

Next we consider the long term effect of the flow upon a given volume. For purposes of illustration we consider the interval from t = 0 to t = 1.2.

```
> tstart := 0; tend := 1.2; deltat := 0.2;
```

In preparation, a slightly different view of the vector field is needed. (In general, adjust the limits in the following command to ensure viewing of an appropriate portion of the vector field.)

```
> Fplot2:= fieldplot( F(x,y), x = -6.5 .. 2.5 , y = 0 .. 9.50,
  arrows=SLIM ):
```

Instead of using 1-step Euler we use Maple's built-in differential equations solver to compute the flow line through each corner of the flowbox.

```
> sys:= diff(x(t),t)=F(x,y)[1], diff(y(t),t)=F(x,y)[2] ;
> unknowns:= {x(t), y(t)};
> init1:= x(0)=a, y(0)=b;
> init2:= x(0)=aplus, y(0)=b;
> init3:= x(0)=a, y(0)=bplus;
> init4:= x(0)=aplus, y(0)=bplus;
> collection:= corners:
> frames:= display( {Fplot2, Testbox} ):
```

Create flow line through each of the four corners of the box

```
> LLcorner := dsolve({sys, init1}, unknowns, type=numeric):
> LRcorner := dsolve({sys, init2}, unknowns, type=numeric):
> ULcorner := dsolve({sys, init3}, unknowns, type=numeric):
> URcorner := dsolve({sys, init4}, unknowns, type=numeric):
> for _t from tstart to tend by deltat do
>   LL := subs( LLcorner(_t), [x(t),y(t)] ):
>   LR := subs( LRcorner(_t), [x(t),y(t)] ):
>   UL := subs( ULcorner(_t), [x(t),y(t)] ):
>   UR := subs( URcorner(_t), [x(t),y(t)] ):
>   frames := frames, display({Fplot2, polygonplot([LL, LR, UR, UL, LL
  ], color=RED)}):
>   collection := collection, [LL,LR,UR,UL,LL];
> od:
```

Animation

Single Frame

```
>
```

What is going on here?
Does this calculation shed any more light (or perhaps cast a shadow) on the 1-step Euler approximation that we did earlier, especially when we took deltat = 0.9?
To explore these concepts further, rework everything (both local and global portions of this worksheet) with a new vector field, say for example, F(x,y) = (y,-x), or F(x,y) = (x,0), F(x,y) = (y,0), or F(x,y) = (x/sqrt(x^2+y^2), y/sqrt(x^2+y^2)). What do you observe in each of these cases?

```
>
>
```