

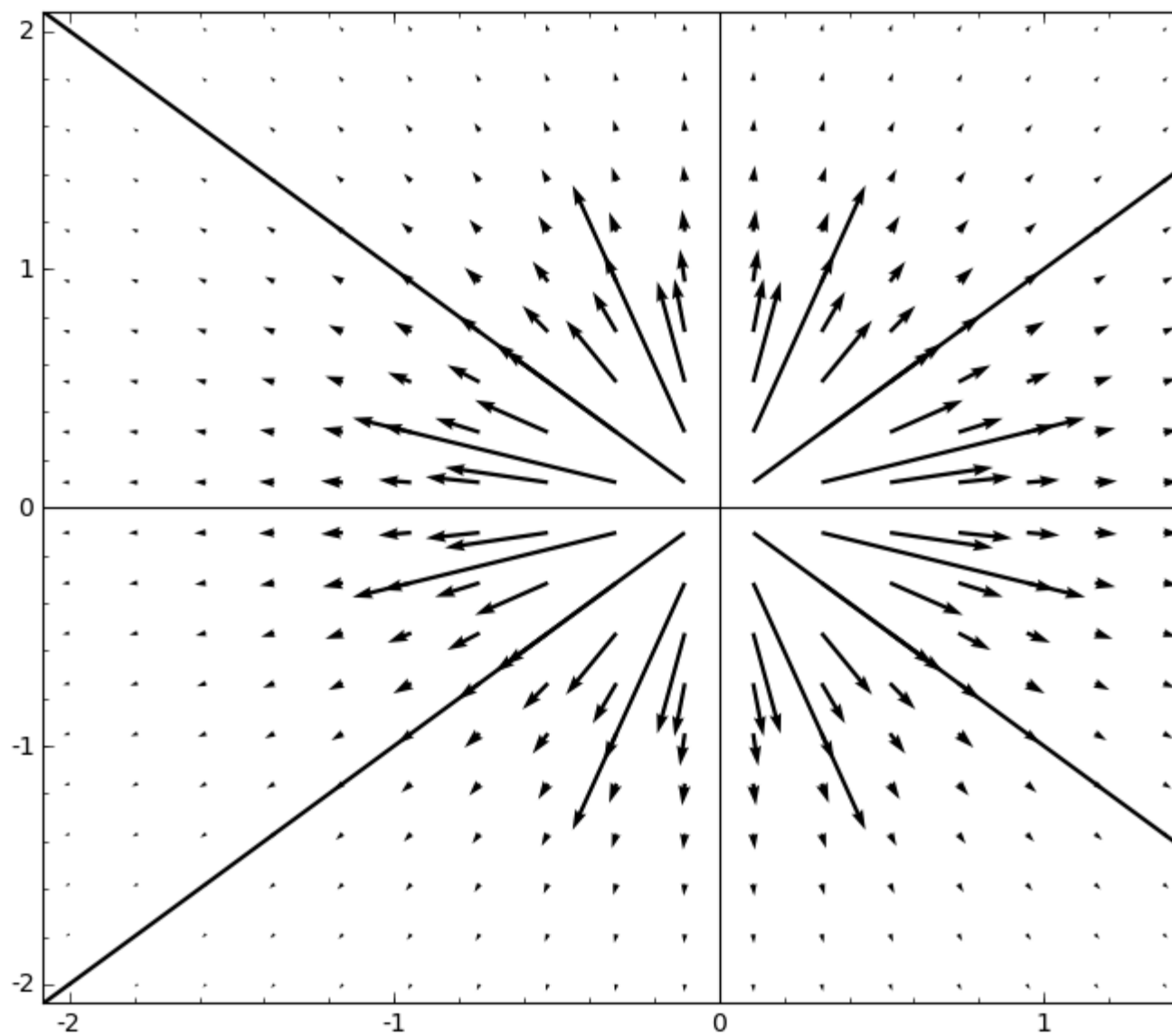
Math 212 April 19

```
# field due to a point charge q at point (a,b), according to Coulomb's
law (scaled)
# Q = [q,(a,b)]
def point_charge(Q,x,y,unit=false):
    q = Q[0]; a = Q[1][0]; b = Q[1][1];
    p = vector([x,y])
    p0 = vector([a,b])
    v = p-p0
    r = sqrt(v*v)
    if unit:
        return v/r
    else:
        return (q/r^3)*v
```

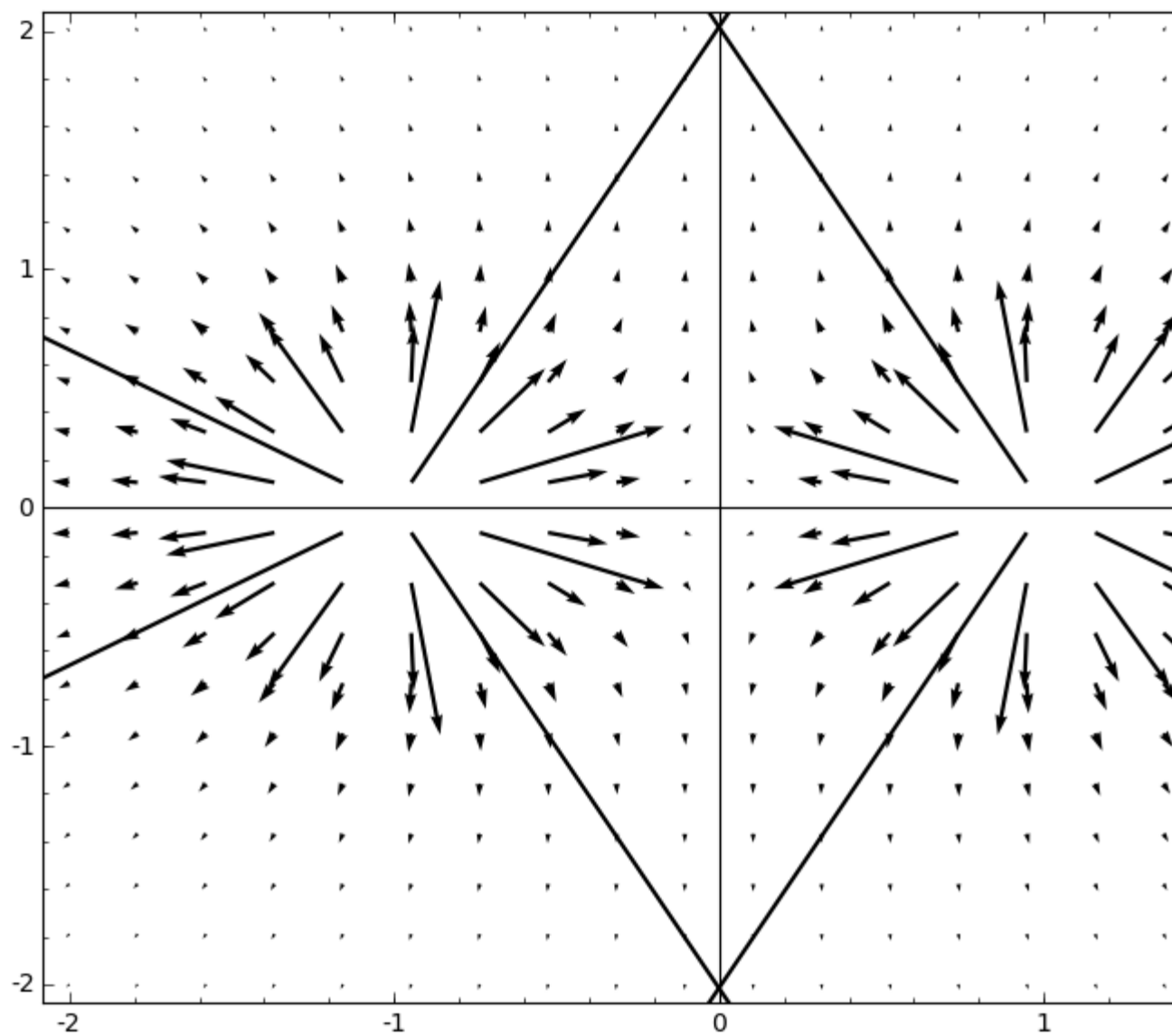
```
# return the field due to a list of charges
def many_charges(L,x,y,unit=false):
    result = vector(RR,(0,0))
    for Q in L:
        result += point_charge(Q,x,y)
    if unit:
        r = sqrt(result*result)
        return result/r
    else:
        return result
```

```
point_charge([1,(0,0)],1,1)
(1/4*sqrt(2), 1/4*sqrt(2))
```

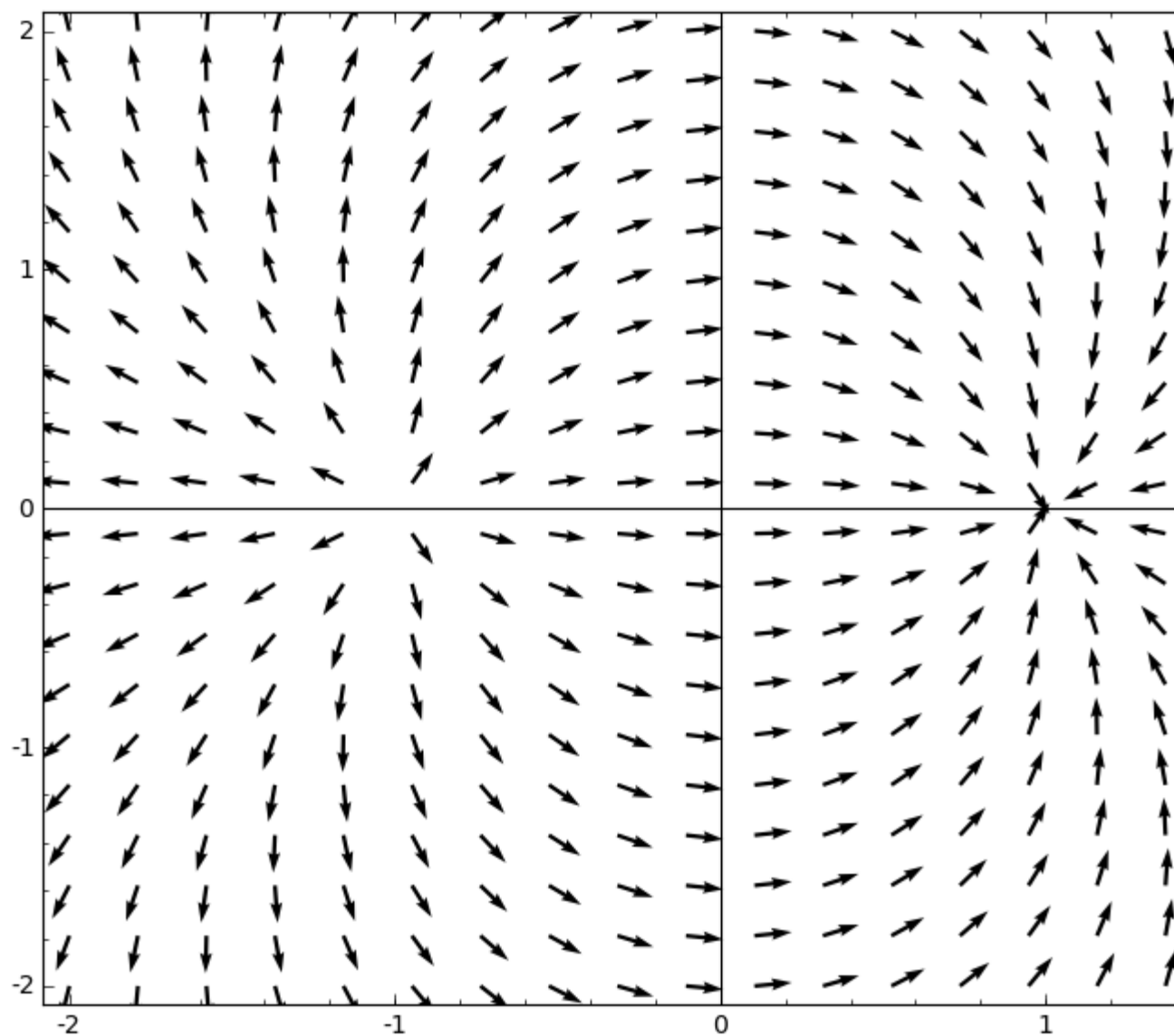
```
var('x y')
plot_vector_field(point_charge([1,(0,0,0)],x,y),(x,-2,2),(y,-2,2))
```



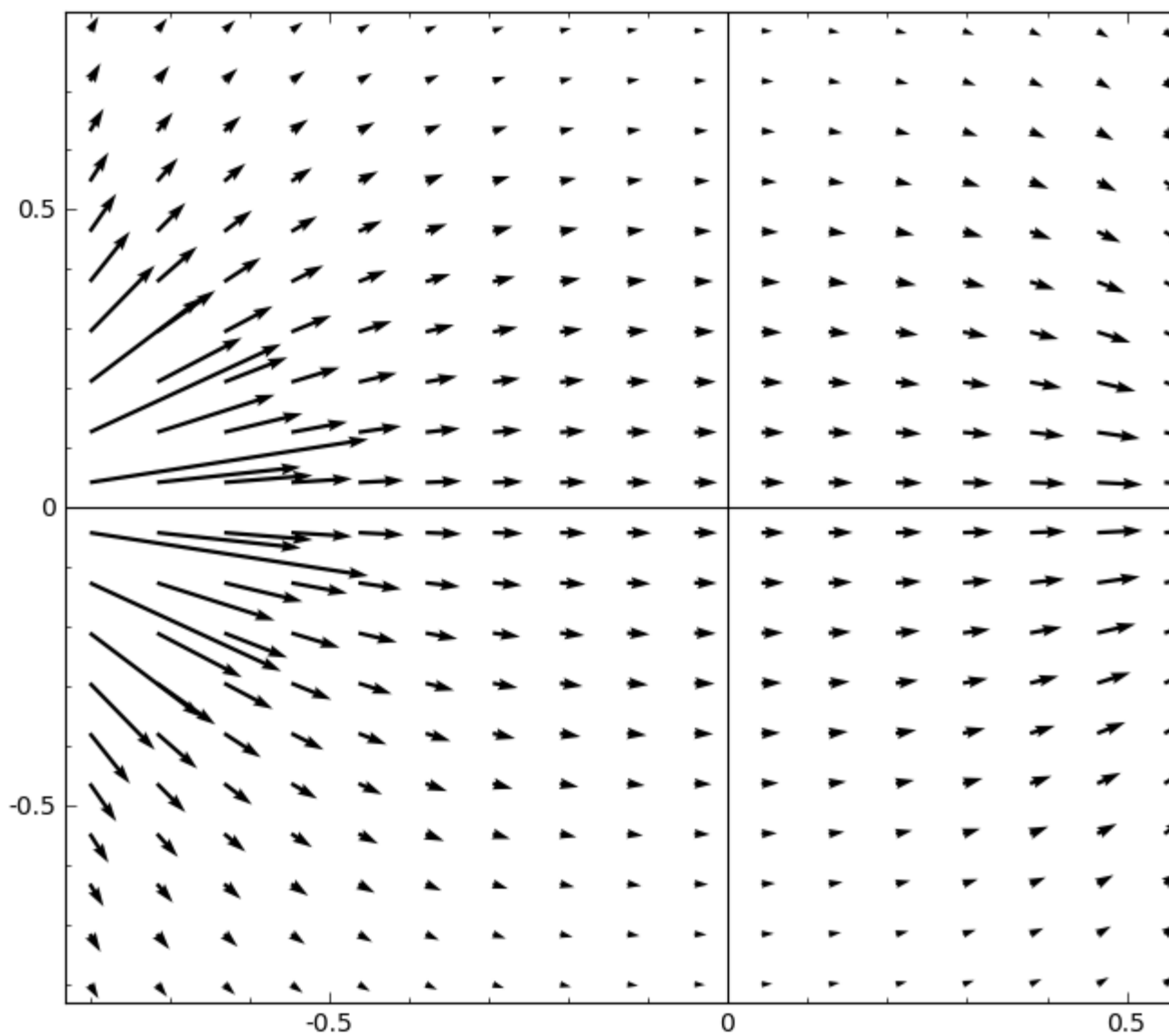
```
L = [[1,(-1,0)],[1,(1,0)]]  
plot_vector_field(many_charges(L,x,y),(x,-2,2),(y,-2,2))
```



```
L = [[1,(-1,0)],[-1,(1,0)]]  
plot_vector_field(many_charges(L,x,y,unit=true),(x,-2,2),(y,-2,2))
```



```
L = [[1,(-1,0)],[-1,(1,0)]]
plot_vector_field(many_charges(L,x,y),(x,-0.8,0.8),(y,-0.8,0.8))
```



```
plot_vector_field?
```

```
L = [[1,(cos(2*pi*t/5).n(),sin(2*pi*t/5).n())] for t in [0,1,2,3,4]]
plot_vector_field(many_charges(L,x,y,true),(x,-2,2),
(y,-2,2),plot_points=25)
```

