Electric Fields of Point Charges

We define several electric fields below. For the sake of visualization I stick to two-dimensional fields but the three dimensional case is not particularly different. To begin I define the Coulomb fields at the origin using "Ec" in the code below. These pictures will give us a sense of the shape of the fields however, to calculate specific magnitudes we'll need to put back in proper units and work things out from the proper vector calculus equations. [I'm using StreamPlot because I wanted to illustrate the field lines and the vector fields in one nice picture. The field lines are actually the "flow" of the vector field. These are so-called integral curves of the field. At any point the tangent to the integral curve reproduces the vector field for which it is an integral curve.]

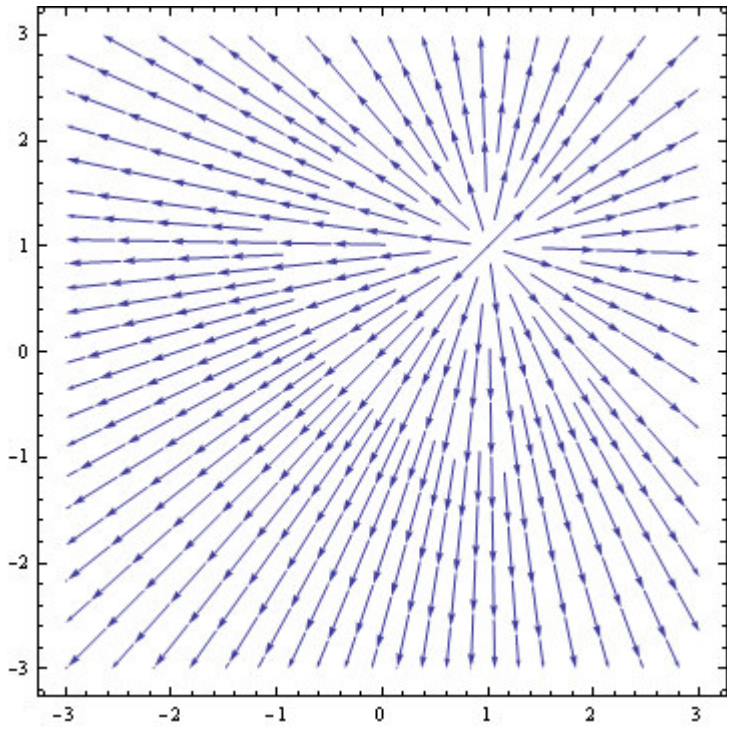
```
Clear[Ec];
```

```
Ec[x_, y_] := Ec[x, y] = {x/(x^2 + y^2)^1.5, y/(x^2 + y^2)^1.5}
```

Monopole Field: aka the Coulomb field:

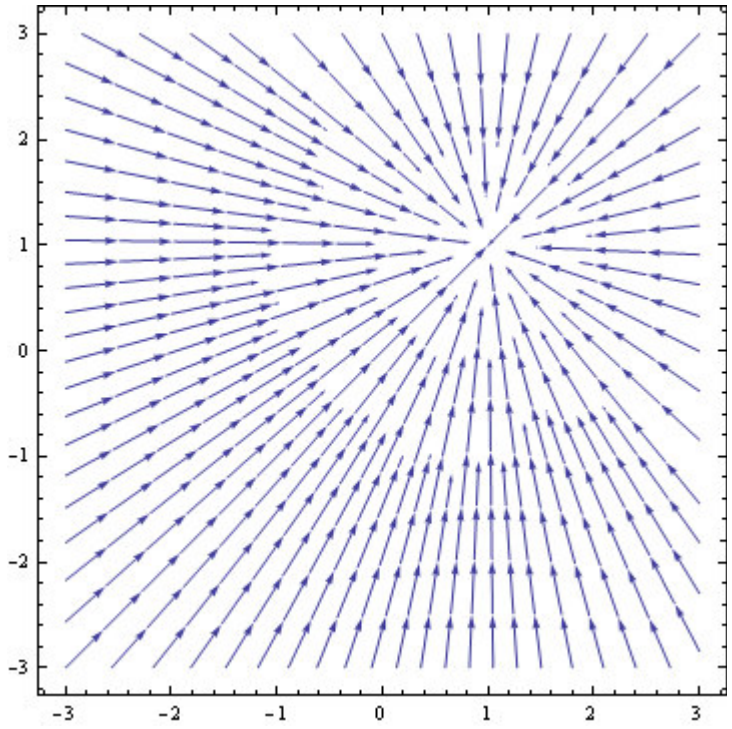
Below we see the electric field lines for a single POSTIVE point charge placed at (1,1).

```
StreamPlot[Ec[x - 1, y - 1], {x, -3, 3}, {y, -3, 3}]
```



Below we see the electric field lines for a single NEGATIVE point charge placed at (1,1).

```
StreamPlot[ -Ec[x - 1, y - 1], {x, -3, 3}, {y, -3, 3}]
```

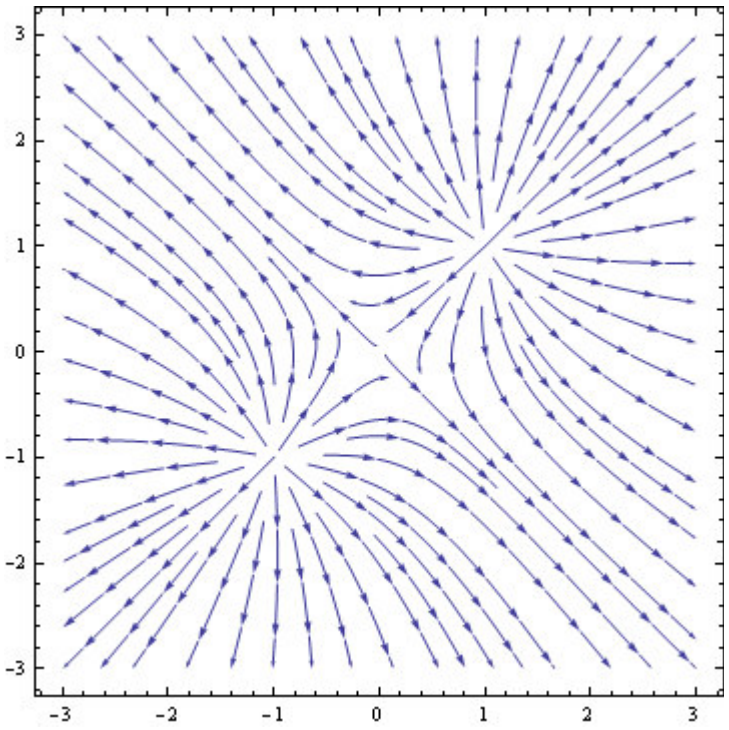


You can imagine placing a little test charge q into these fields. The postive charges' field would push the q away from (1,1) whereas the negative charges' field would pull the test charge in towards (1,1).

Fields due to two point charges:

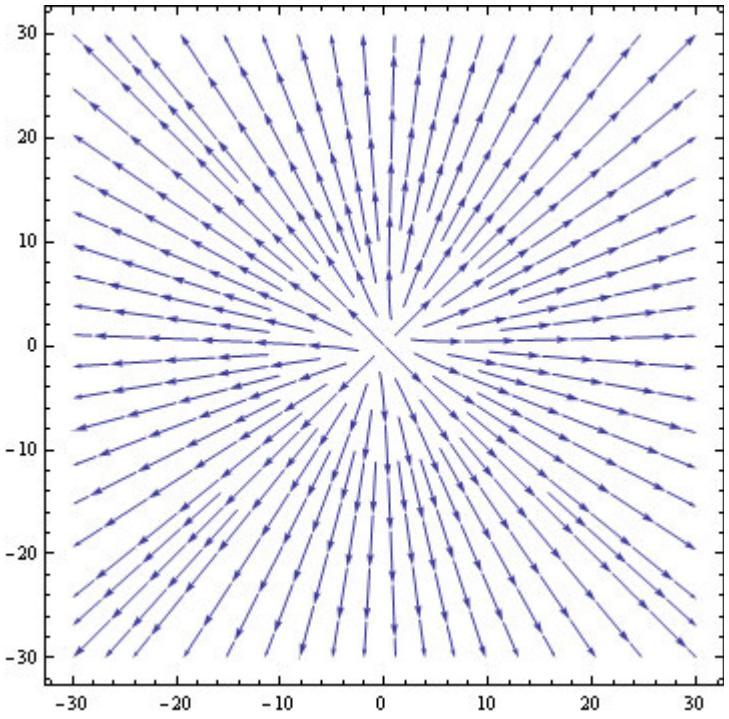
Suppose we have two positive charges, one at (1,1) and another at (-1,-1). The net electric field will be formed from the superposition of the two fields: E(x,y) = Ec(x-1,x-1)+Ec(x+1,x+1). This fact follows from the definition of the electric field as well as Newton's law that the net force is the sum of the forces.

```
StreamPlot[ Ec[x + 1, y + 1] + Ec[x - 1, y - 1], {x, -3, 3}, {y, -3, 3}]
```



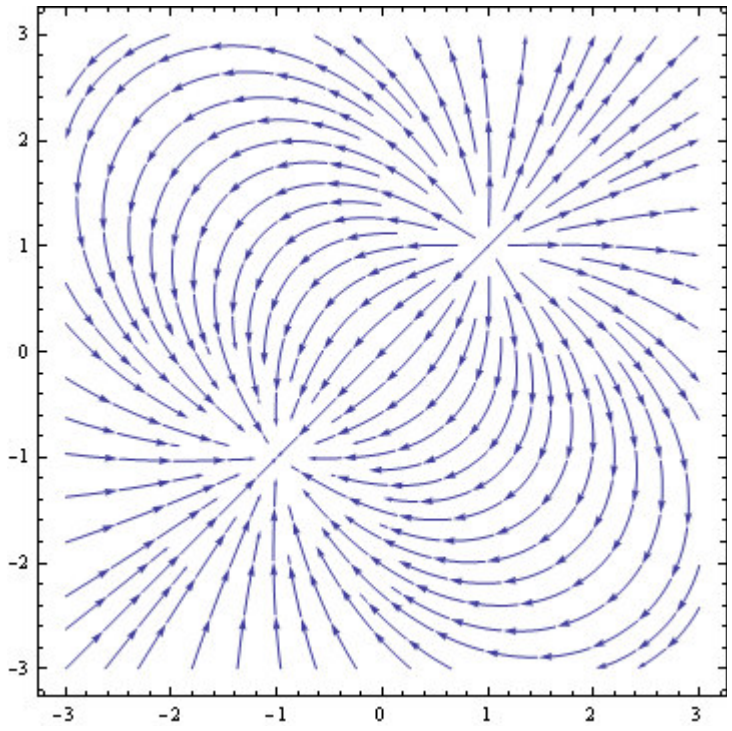
Now, this appears a little different than what we saw before. However, suppose we zoom out to the scale of about 10:

```
StreamPlot[ Ec[x + 1, y + 1] + Ec[x - 1, y - 1], {x, -30, 30}, {y, -30, 30}]
```



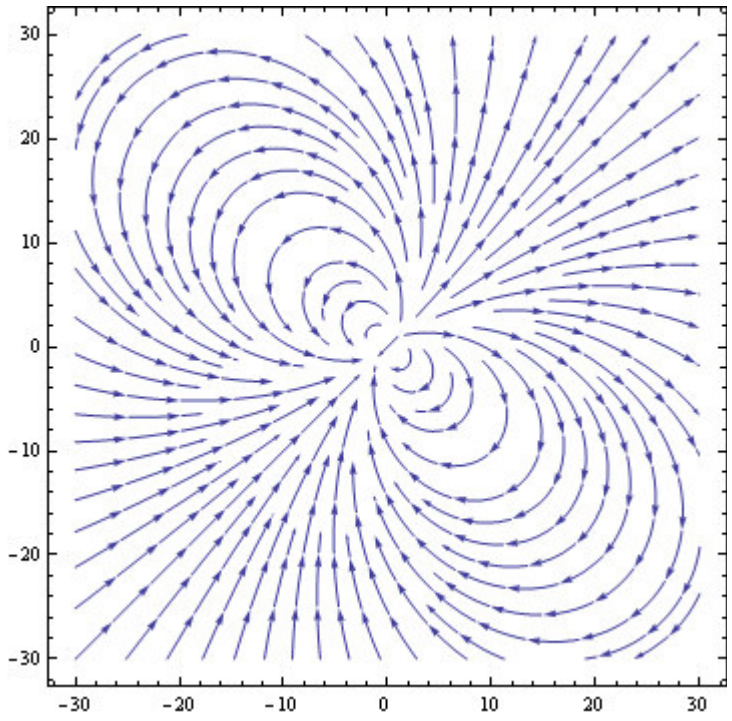
The field above appears to be the Coulomb field from this scale. It's as if we just had two point charges sitting near the origin. What happens if the charges are of different polarity? Consider a POSITIVE charge at (1,1) and a negative charge at (-1,-1). The net field is E(x,y) = Ec(x-1,x-1)-E(x+1,x+1), beginning with the scale we were using:

```
StreamPlot[ -Ec[x + 1, y + 1] + Ec[x - 1, y - 1], {x, -3, 3}, {y, -3, 3}]
```



Zooming out doesn't change too much in terms of the big pattern. Take the same field as above viewed on the scale of 10,

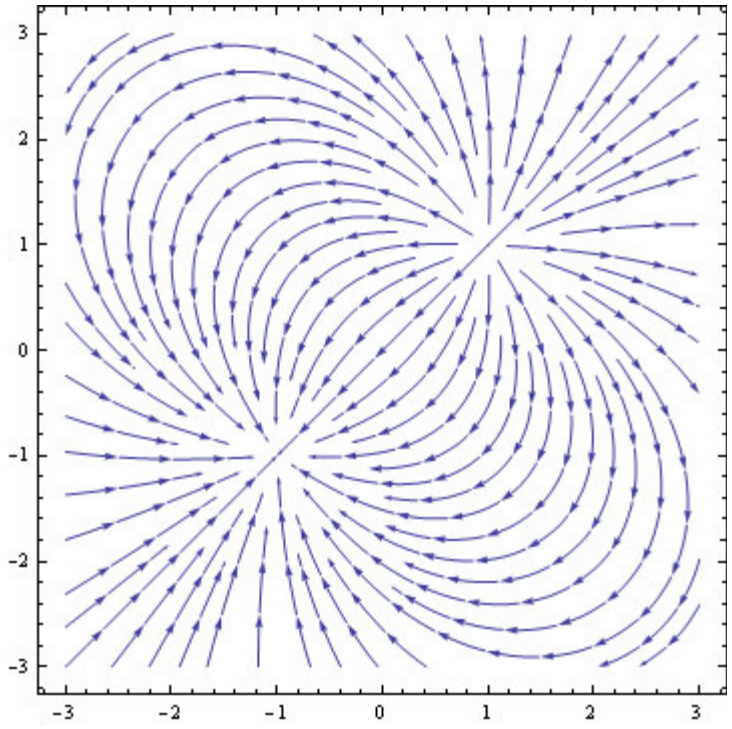
```
StreamPlot[ -Ec[x + 1, y + 1] + Ec[x - 1, y - 1], {x, -30, 30}, {y, -30, 30}]
```



This is a DIPOLE field. Its magnitude is much weaker than the Coulomb field for large distances. Or more precisely, the strength of a

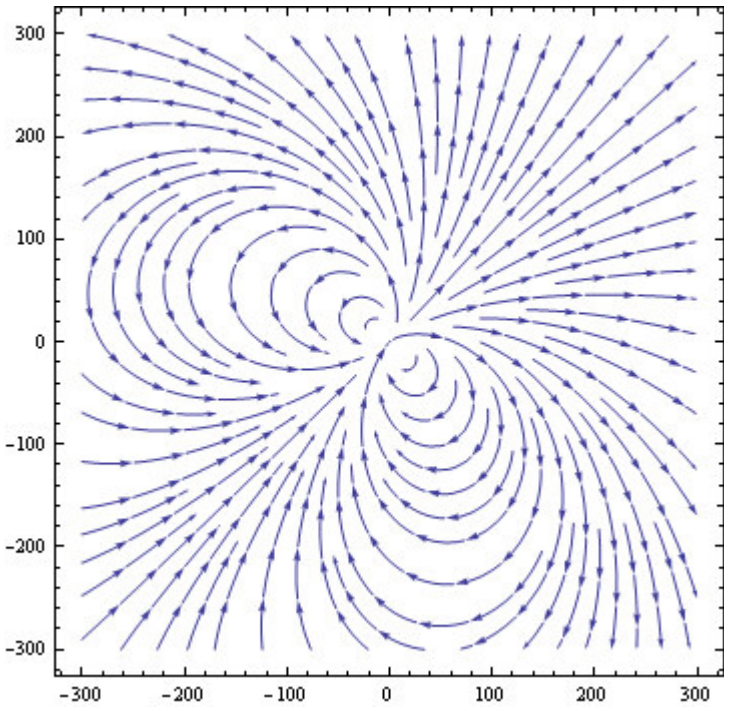
given dipole field drops according to $\frac{1}{r^3}$ in a particular direction. In contrast the Coulomb field drops according the $\frac{1}{r^2}$ factor in Coulombs Law. Note that this is only the case if the postive and negative charges are of exactly equal strength. If either charge is just a little bit bigger we'll again get a Coulomb field for large distances. For example, suppose we have a POSITIVE charge of 1.01 units,

```
StreamPlot[1.01*Ec[x - 1, y - 1] - Ec[x + 1, y + 1], {x, -3, 3},
 {y, -3, 3}]
```



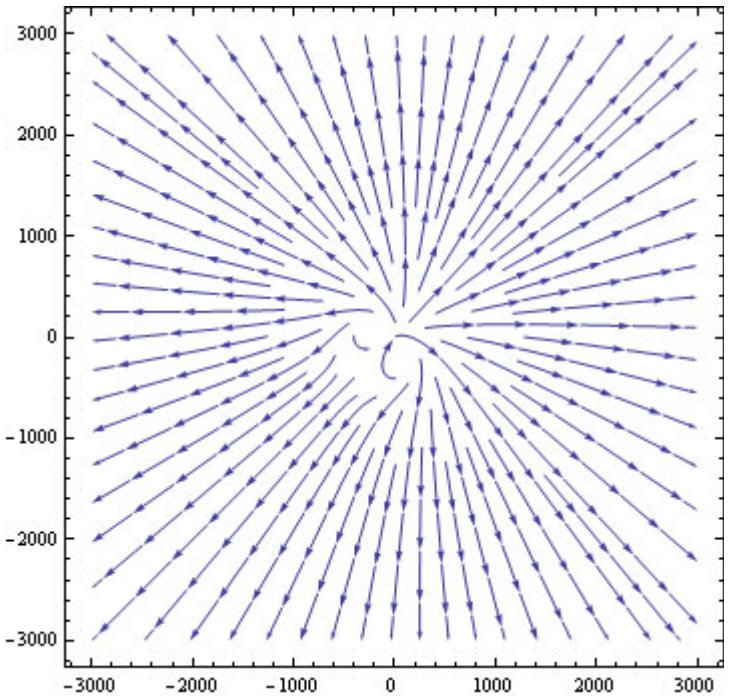
You're probably saying, that's just the dipole field. Well, ok, let's zoom out to the 100 scale and see what we see:

```
StreamPlot[1.01*Ec[x - 1, y - 1] - Ec[x + 1, y + 1], {x, -300, 300},
 {y, -300, 300}]
```



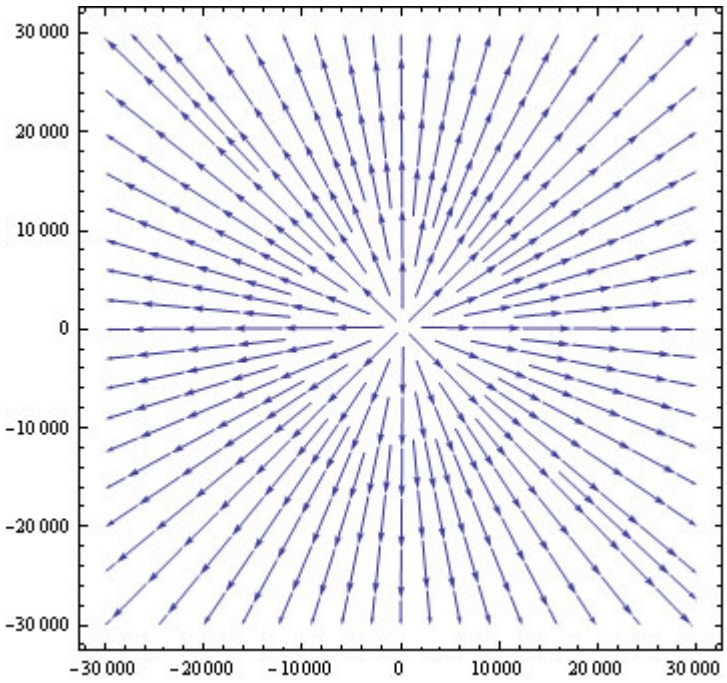
Ok, have patience, the 1000 scale,

```
StreamPlot[1.01*Ec[x - 1, y - 1] - Ec[x + 1, y + 1], {x, -3000, 3000},
 {y, -3000, 3000}]
```



See. What'd I say? Fine, the 10,000 scale:

```
StreamPlot[ 1.01 * Ec[x - 1, y - 1] - Ec[x + 1, y + 1], {x, -30 000, 30 000},
 {y, -30 000, 30 000}]
```

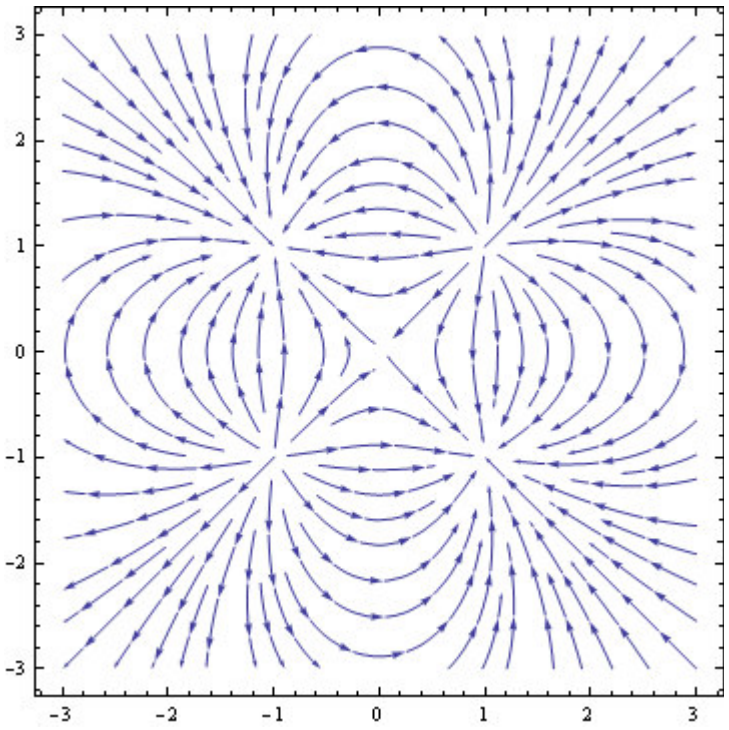


Multiple charges in the plane: superposition

To find the net electric field we need only identify all the point charges in the system and add together all the Coulomb fields. Adding vector fields together again gives vector fields. Technically these vector fields have domains which fail to exist at the place where the charges are located. Charges occur at singularities in the fields. When we add together several fields there will be more singular points for the composite field (unless of course the charges are all in the same place). The resulting field can be rather complicated. However, if the net charge works out to zero then the large-scale field will not be a Coulomb field. Instead you could get a dipole field pattern, or a quadrapole field etc...

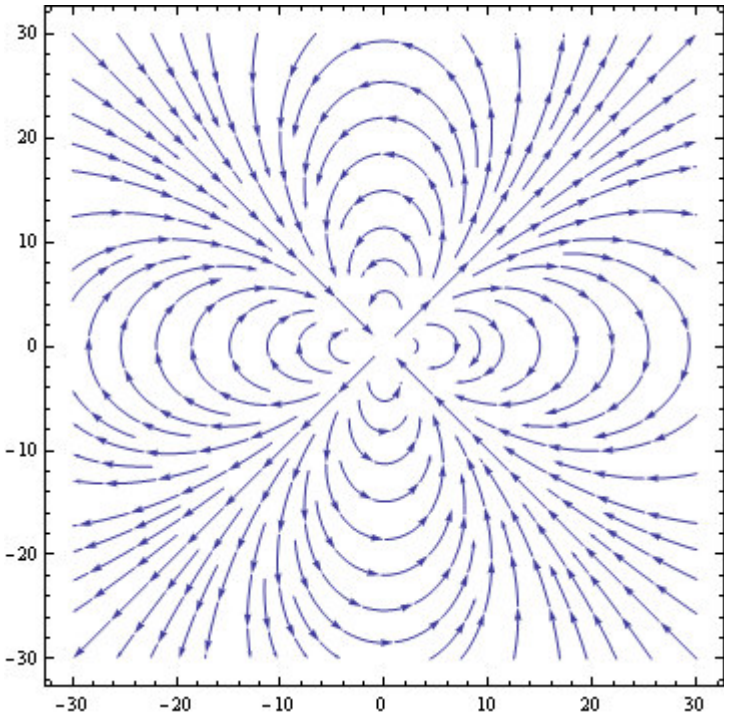
In the example below I put two positive charges and two negative charges at opposite corners of a square centered at the origin.

```
StreamPlot[
 Ec[x - 1, y - 1] + Ec[x + 1, y + 1] -  Ec[x + 1, y - 1] - Ec[x - 1, y + 1],
 {x, -3, 3}, {y, -3, 3}]
```



Zooming out by a factor of 10, you get a nice picture of the quadrapole field:

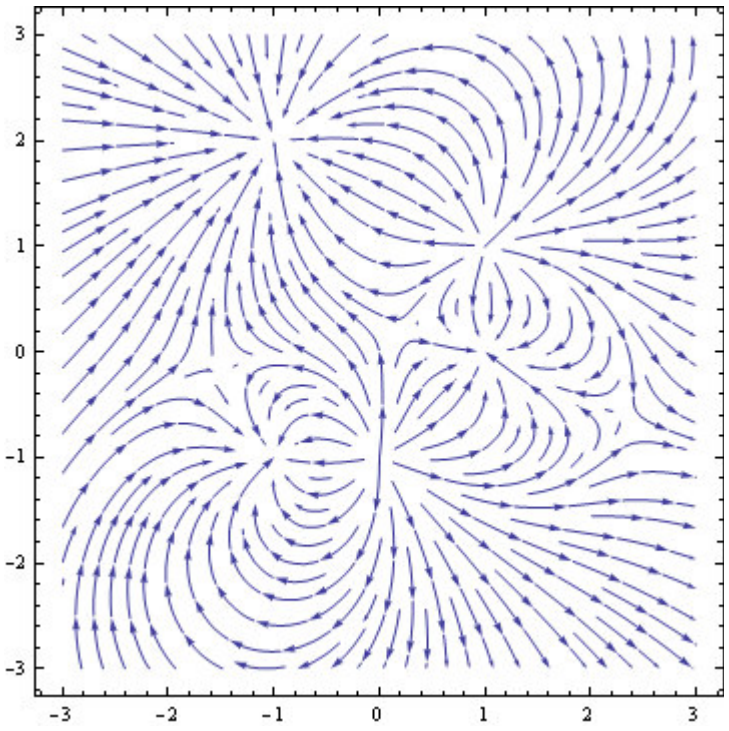
```
StreamPlot[
 Ec[x - 1, y - 1] + Ec[x + 1, y + 1] -  Ec[x + 1, y - 1] - Ec[x - 1, y + 1],
 {x, -30, 30}, {y, -30, 30}]
```



Naturally, we don't always have a net zero charge for a system. Often we are most interested in the field local to the charges. The local field has all sorts of bends and quirks. Only for large scales can I make general sweeping statements about how the charge

makes the fields look. Locally we have to deal with the actual precise physical location of charge. For example, in the field below we see how charges at (1,1),(1,0),(-1,2), (-1,-1) and (0,-1) set up a rather complicated field near the origin.

```
StreamPlot[
 2 * Ec[x - 1, y - 1] - Ec[x - 1, y] -  3 * Ec[x + 1, y - 2]
                    - Ec[x + 1, y + 1] + 2 * Ec[x, y + 1], {x, -3, 3}, {y, -3, 3}, StreamPoints → Fine]
```



Imagine placing a test charge in that mess. Which way would it go? For fun try taking this code and modifying my charge strengths to make other spagetti electric dishes.

I wrote this notebook to help us visualize two-dimensional electrostatics. These pictures don't help be obtain quantitative data but I do think they increase our qualitative understanding of how electric fields combine. The next theoretical question to think about is how to find electric fields from distributions of charges. In other words, what if we take the point charges and smear them over some shape? What will be the electric field from such a distribution. I'm not sure I have sufficient *Mathematica* skillz to attempt that in here. One last demo: the three dimensional Coulomb and Dipole fields:

In[11]:=

```
Clear[E3d];
```

In[16]:=

```
E3d[x_ , y_ , z_ ] := E3d[x, y, z] =
  { x / (x^2 + y^2 + z^2)^1.5 , y / (x^2 + y^2 + z^2)^1.5,
   z / (x^2 + y^2 + z^2)^1.5 }
```

I have trouble with magnitude plotting. To fix this I made it bigger, technically this is not the Coulomb field unless you set p=0. [just a parameter to see direction of Coulomb field easier]

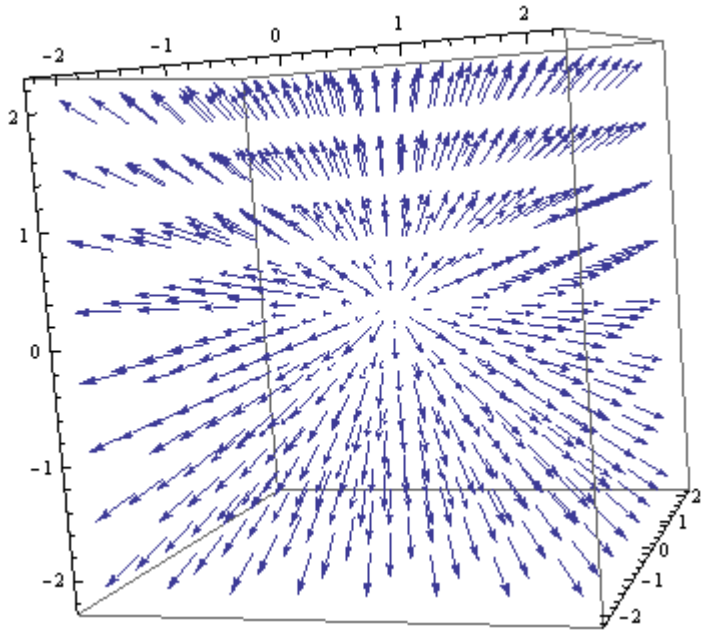
In[28]:=

```
p := 1
```

In[29]:=

```
VectorPlot3D[(x^2 + y^2 + z^2)^p * E3d[x, y, z], {x, -2, 2},
 {y, -2, 2}, {z, -2, 2}, VectorScale → Small]
```

Out[29]=



The dipole field again stems from two charges of equal but opposite parity:

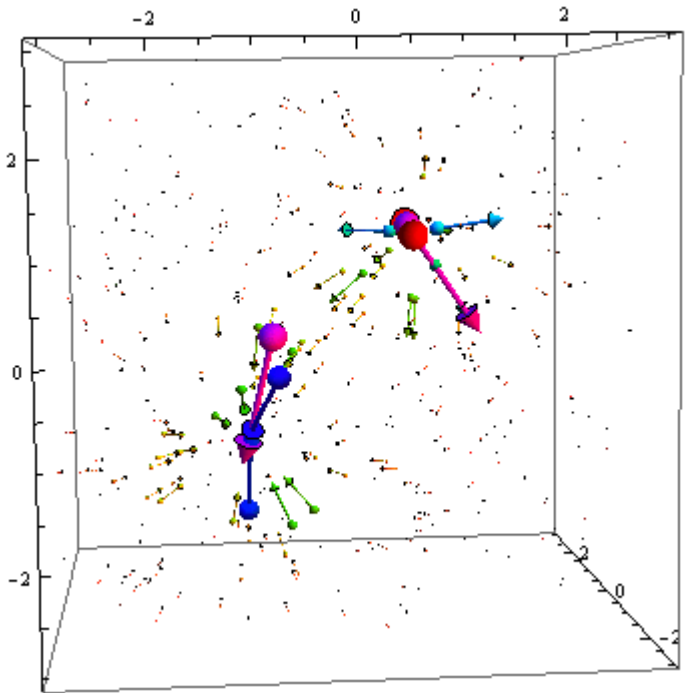
In[75]:=

```
points = RandomReal[{-3, 3}, {400, 3}];
```

In[77]:=

```
VectorPlot3D[
 E3d[x - 1, y - 1, z - 1] -
  E3d[x + 1, y + 1, z + 1], {x, -3, 3}, {y, -3, 3}, {z, -3, 3},
 VectorPoints → points, PlotRange → All,
 VectorStyle →
  {{Graphics3D[Sphere[{0, 0, 0}, 0.5]],
    Graphics3D[Cone[{{0, 0, 0}, {1, 0, 0}}, 0.5]]}},
 VectorScale → {Automatic, Scaled[0.33]}, VectorColorFunction → Hue]
```

Out[77]=



You can tinker with the graphic options, maybe you'll get a better picture than I did. In any event, we will soon learn about the electric potential and the problem of visualizing vector fields will be replaced with the easier task of finding level curves or surfaces. *Mathematica* seems to do better with level surfaces that with 3D vector fields. Of course, I have only tinkered about an hour. I'm sure if you spend a few days you could make way better pictures.

In[44]:

```
? VectorPlot3D
```

"\!\(*RowBox[{\"VectorPlot3D\", \"[\", RowBox[{RowBox[{\"{\", RowBox[{SubscriptBox[StyleBox[\"v\", \"TI\"], StyleBox[\"x\", \"TI\"]], \",\", SubscriptBox[StyleBox[\"v\", \"TI\"], StyleBox[\"y\", \"TI\"]], \",\", SubscriptBox[StyleBox[\"v\", \"TI\"], StyleBox[\"z\", \"TI\"]]}], \"}\"}], \",\", RowBox[{\"{\", RowBox[{StyleBox[\"x\", \"TI\"], \",\", SubscriptBox[StyleBox[\"x\", \"TI\"], StyleBox[\"min\", \"TI\"]], \",\", SubscriptBox[StyleBox[\"x\", \"TI\"], StyleBox[\"max\", \"TI\"]]}], \"}\"}], \",\", RowBox[{\"{\", RowBox[{StyleBox[\"y\", \"TI\"], \",\", SubscriptBox[StyleBox[\"y\", \"TI\"], StyleBox[\"min\", \"TI\"]], \",\", SubscriptBox[StyleBox[\"y\", \"TI\"], StyleBox[\"max\", \"TI\"]]}], \"}\"}], \",\", RowBox[{\"{\", RowBox[{StyleBox[\"z\", \"TI\"], \",\", SubscriptBox[StyleBox[\"z\", \"TI\"], StyleBox[\"min\", \"TI\"]], \",\", SubscriptBox[StyleBox[\"z\", \"TI\"], StyleBox[\"max\", \"TI\"]]}], \"}\"}]}], \"]\"}]\) generates a 3D vector plot of the vector field \!\(*RowBox[{\"{\", RowBox[{SubscriptBox[StyleBox[\"v\", \"TI\"], StyleBox[\"x\", \"TI\"]], \",\", SubscriptBox[StyleBox[\"v\", \"TI\"], StyleBox[\"y\", \"TI\"]], \",\", SubscriptBox[StyleBox[\"v\", \"TI\"], StyleBox[\"z\", \"TI\"]]}], \"}\"}]\) as a function of \!\(*StyleBox[\"x\", \"TI\"]\), \!\(*StyleBox[\"y\", \"TI\"]\) and \!\(*StyleBox[\"z\", \"TI\"]\).\n\!\(*RowBox[{\"VectorPlot3D\", \"[\", RowBox[{RowBox[{\"{\", RowBox[{SubscriptBox[StyleBox[\"field\", \"TI\"], StyleBox[\"1\", \"TR\"]], \",\", SubscriptBox[StyleBox[\"field\", \"TI\"], StyleBox[\"2\", \"TR\"]], \",\", StyleBox[\"...\", \"TR\"]}], \"}\"}], \",\", RowBox[{\"{\", RowBox[{StyleBox[\"x\", \"TI\"], \",\", SubscriptBox[StyleBox[\"x\", \"TI\"], StyleBox[\"min\", \"TI\"]], \",\", SubscriptBox[StyleBox[\"x\", \"TI\"], StyleBox[\"max\", \"TI\"]]}], \"}\"}], \",\", RowBox[{\"{\", RowBox[{StyleBox[\"y\", \"TI\"], \",\", SubscriptBox[StyleBox[\"y\", \"TI\"], StyleBox[\"min\", \"TI\"]], \",\", SubscriptBox[StyleBox[\"y\", \"TI\"], StyleBox[\"max\", \"TI\"]]}], \"}\"}], \",\", RowBox[{\"{\", RowBox[{StyleBox[\"z\", \"TI\"], \",\", SubscriptBox[StyleBox[\"z\", \"TI\"], StyleBox[\"min\", \"TI\"]], \",\", SubscriptBox[StyleBox[\"z\", \"TI\"], StyleBox[\"max\", \"TI\"]]}], \"}\"}]}], \"]\"}]\) plots several vector fields." >>

Created with Wolfram *Mathematica* 7.0