In[1]:=
```
(*MATHEMATICA companion to Lab 201: Field from point charges*)
```

In[2]:=
```
(*0. Commands and graphing objects used:
     Table
        Graphics
        Arrow - to plot vectors
        Point
        Disk
        Line
        Show - to combine plots and pictures
        Export
        Norm - length of a vector
        VectorPlot (plots directional field)
        StreamPlot - plots field lines
        FullSimplify[...,Assumptions→
            x ∈ Reals] - to simplify ... assuming x is real (not complex)
    %/.{a→..,b→ ..}- the replacement command;
mostly used when symbolic parameters a,
b, ... in the previous expression (%) are to be replaced by speciific values
   [[1]] - the 1st element of a list
   /@ - advanced command,
which indicates that the operation is applied to every element of the list. Useful for
 graphing multiple objects. E.g. if 'list' is a long list of {x,y} coordinates,
the command Graphics[Point /@ list] will show an array of
 points with indicated coordinates. NOT CRUCIAL at this
 stage.(Also, see files IntroToMathematicaI and II on our web page https://
    web.njit.edu/~vitaly/121/ *)
```

In[3]:=
```
(*Units: later we will learn to carry out transformations of equations and computations
  for variables with units. At the moment, we will use only SI base units*)
```

In[4]:=
```
repSI := {ke → 9. × 10^9, nC → 10.^-9}
```

In[5]:= (*I. Field from a single charge: $\vec{E} = k_e q \vec{r} / r^3$*)

In[6]:=
```
efield[rvec_, q_] := ke q rvec / Norm[rvec]^3
```

In[7]:=
```
rvec := {x, y}
```

In[8]:=
```
efield[rvec, 1 nC] /. repSI
```

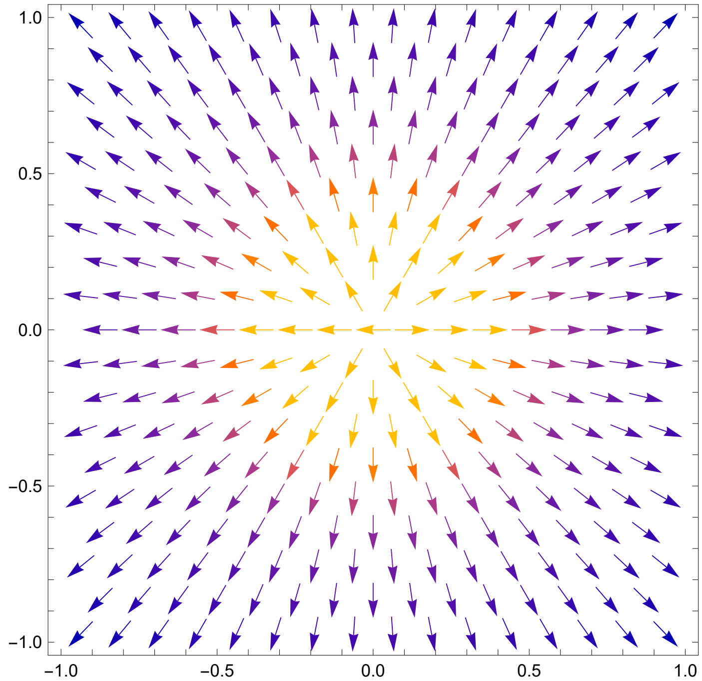
Out[8]= $\left\{\frac{9.\, x}{\left(\text{Abs}[x]^2+\text{Abs}[y]^2\right)^{3/2}}, \frac{9.\, y}{\left(\text{Abs}[x]^2+\text{Abs}[y]^2\right)^{3/2}}\right\}$

In[9]:=
```
ef = FullSimplify[%, Assumptions → {x ∈ Reals, y ∈ Reals}]
```

Out[9]= $\left\{\frac{9.\, x}{\left(x^2+y^2\right)^{3/2}}, \frac{9.\, y}{\left(x^2+y^2\right)^{3/2}}\right\}$

In[10]:=
```
fig1P = VectorPlot[ef, {x, -1, 1}, {y, -1, 1}]
```

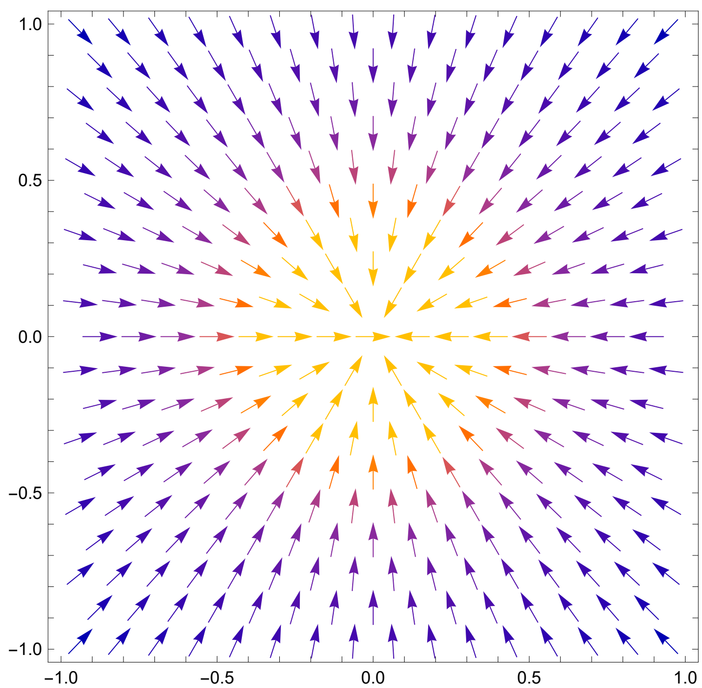
Out[10]=



In[11]:=
```
(*and for negative charge of -1 nC:*)
```

In[12]:=
```
fig1N = VectorPlot[-ef, {x, -1, 1}, {y, -1, 1}]
```

Out[12]=



In[13]:=
```
(*Note: Mathematica gives all vectors the same length and indicates their magnitude
  by color. Why?? It would very hard to assign vectors proportional lenths
     due to strong singularity at r→0. Let us try by "hand" as described
     below (this is an advanced part which can be followed superficially)*)
```

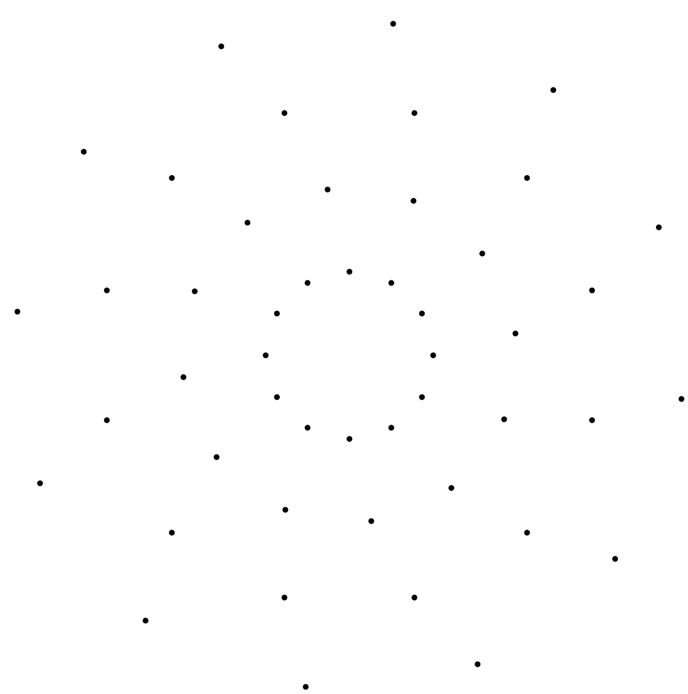
In[14]:=
```
(*first,
   I define a list of points from which to start vectors. To comply with symmetry,
   I arrange the points in 4 circles (12 points in each)with radii 1,2,3,4). Each
 circle is rotated by 1/4 of a notch to minimize overlap of future
 vectors. I use the real and imaginary parts
 of a complex number to locate a point,
but basically this is the same as cos and sin in polar coordinates *)
```

In[15]:=
```
list1 = Table[{Re[Exp[2 Pi ×
      I n / 12]], Im[Exp[2 Pi I n / 12]]}, {n, 0, 11}];
list2 = Table[{2 Re[Exp[2 Pi ×
       I n / 12]], 2 Im[Exp[2 Pi I n / 12]]}, {n, -3 / 4, 11}];
list3 = Table[{3 Re[Exp[2 Pi ×
       I n / 12]], 3 Im[Exp[2 Pi I n / 12]]}, {n, -1 / 2, 11}];
list4 = Table[{4 Re[Exp[2 Pi ×
       I n / 12]], 4 Im[Exp[2 Pi I n / 12]]}, {n, -1 / 4, 11}];
```

In[16]:=
```
(*now let's have a look:*)
```

In[17]:=
```
Graphics[Point /@ {list1, list2, list3, list4}]
```

Out[17]=



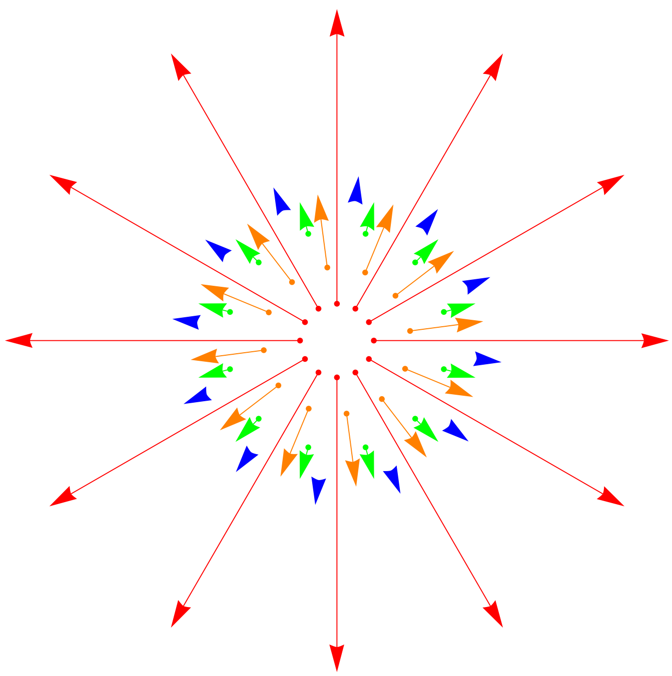
In[18]:=
```
(*now I assign vectors to each point;
scaling of lengths is arbitrary and is given by parameter a. Note that Arrow ("vector")
  depends on 2 pairs of coordinates - beginning and end- hense the Transpose command*)
```

In[19]:=
```
a = .5;
arlist1 = Transpose[{list1, (1 + a * 16) list1}];
arlist2 = Transpose[{list2, (1 + a * 1 / 2 * 4) list2}];
arlist3 = Transpose[{list3, (1 + a * 1 / 3 * 16 / 9) list3}];
arlist4 = Transpose[{list4, (1 + 1 / 4 * a) list4}];
```

In[20]:=
```
(*now I add colors to distinguish the origin of vectors:*)
```

In[21]:=
```
Graphics[{Red, Point /@ list1, Arrow /@ arlist1, Orange, Point /@ list2, Arrow /@ arlist2,
  Green, Point /@ list3, Arrow /@ arlist3, Blue, Point /@ list4, Arrow /@ arlist4}]
```

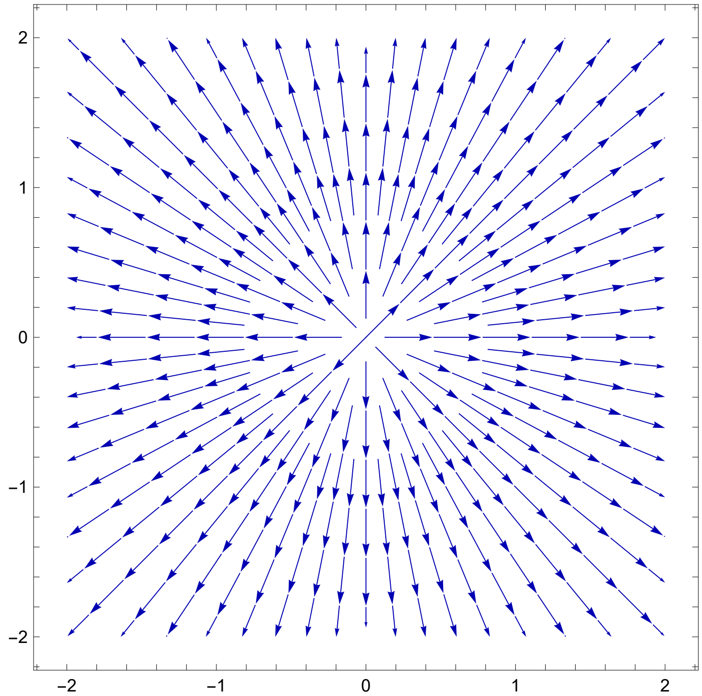
Out[21]=



In[22]:=
```
(*Note the difficulty of plotting scaled vectors due to singularity at r=
 0. Things get even vorse for a negative charge or for a system of charges.
   Thus NEED field lines*)
```

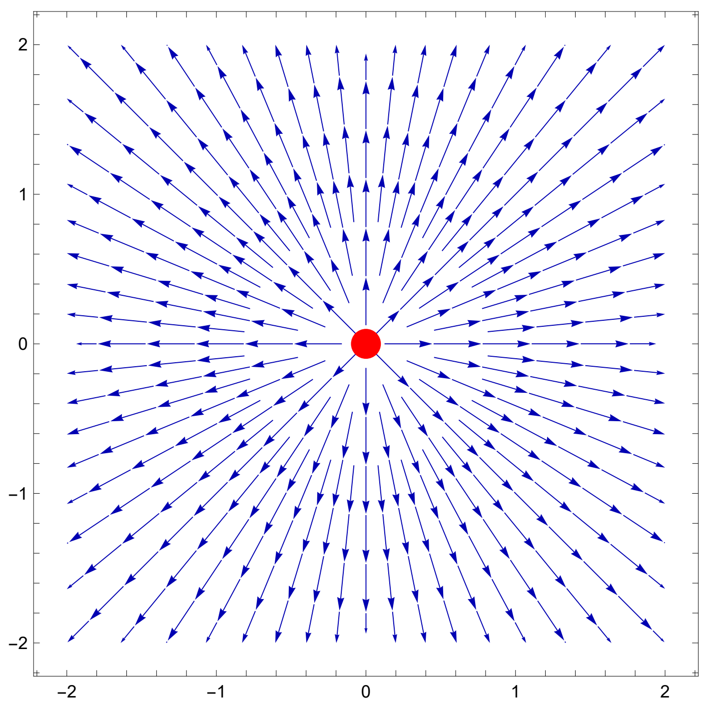
In[23]:=
```
StreamPlot[ef, {x, -2, 2}, {y, -2, 2}]
```

Out[23]=



In[24]:=
```
fig1Pstream = Show[%, Graphics[{Red, Disk[{0, 0}, .1]}]]
```

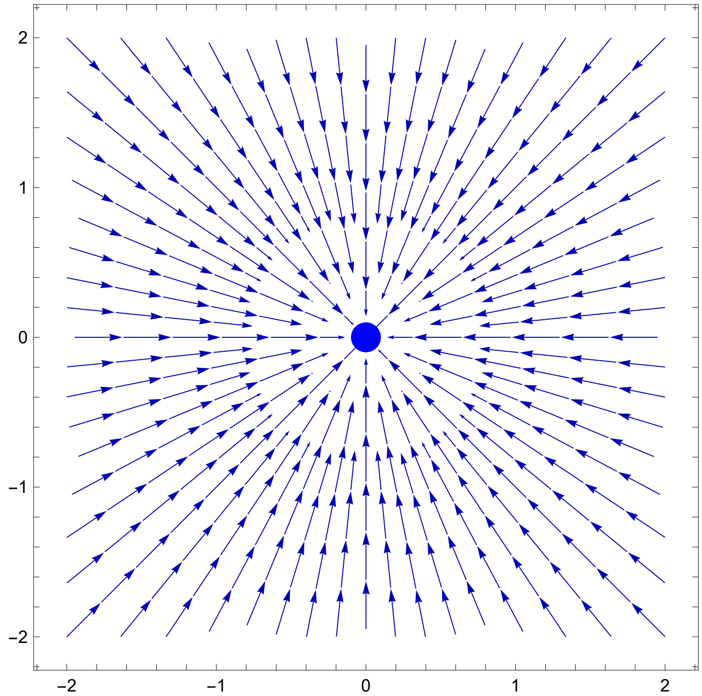
Out[24]=



In[25]:=
```
(*much better; now the same for negative charge*)StreamPlot[-ef, {x, -2, 2}, {y, -2, 2}];
```

In[26]:=
```
fig1Nstream = Show[%, Graphics[{Blue, Disk[{0, 0}, .1]}]]
```

Out[26]=



In[27]:=
```
(*Two charges: field on x-axis, as in Fig.4 of the Manual *)
```

In[28]:=
```
Clear[a, x, y, r1, r2]; a = 1; y = 0; r1 = {-a, 0}; r2 = {a, 0}; q1 = q2 = 1 nC;
```

In[29]:=
```
efield[rvec - r1, q1] + efield[rvec - r2, q2] /. y → 0
```
(* which is $k_e q * \left(\frac{x-1}{|x-1|^3} + \frac{x+1}{|x+1|^3}\right)$

Out[29]=

$$\left\{\frac{\text{ke nC } (-1+x)}{\text{Abs}[-1+x]^3} + \frac{\text{ke nC } (1+x)}{\text{Abs}[1+x]^3}, 0\right\}$$

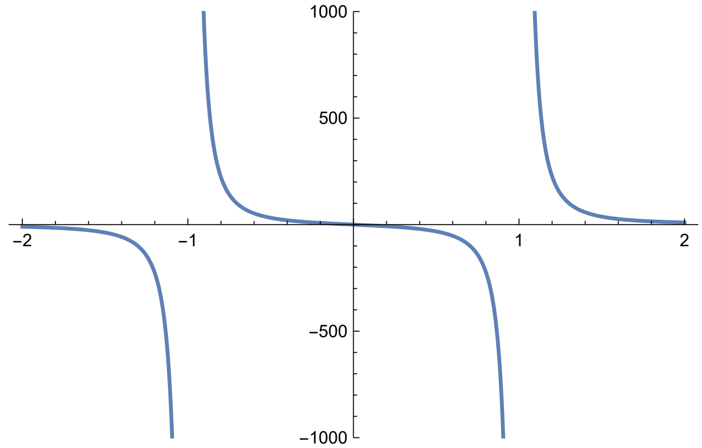
In[30]:=
```
ef2P = %[[1]] /. repSI
```

Out[30]=

$$\frac{9.\ (-1+x)}{\text{Abs}[-1+x]^3} + \frac{9.\ (1+x)}{\text{Abs}[1+x]^3}$$

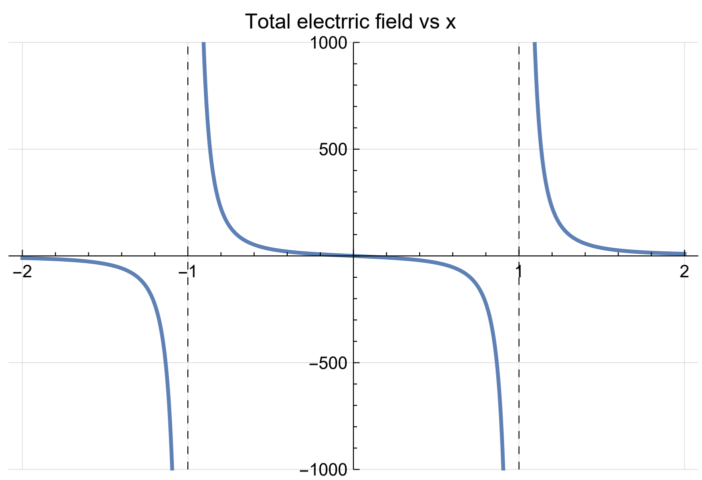
In[31]:=
```
Plot[ef2P, {x, -2, 2}, PlotRange → {-1000, 1000}]
```

Out[31]=



In[32]:=
```
Show[%,
 Graphics[{Dashed, Line[{{-1, -1000}, {-1, 1000}}], Line[{{1, -1000}, {1, 1000}}]}],
 GridLines → Automatic, PlotLabel → "Total electrric field vs x "]
```

Out[32]=



In[33]:=
```
figef2P = %;
```

In[34]:=
```
(*Note enormous increase near charges!!*)
```

In[35]:=

- *(*answer QUESTIONS 1– 4 on pp.23,24 and replot the graph for a dipole with q2=–1 nC*)*

```
(*Part II. Field in 2D - see fig.5 in the Manual*)
```

In[36]:=
```
Clear[a, x, y, r1, r2]; a = 1; r1 = {0, a}; r2 = {0, -a}; q1 = q2 = 1 nC;
```

In[37]:= **efield[rvec - r1, q1] + efield[rvec - r2, q2]**

Out[37]=

$$\left\{\frac{\text{ke nC x}}{\left(\text{Abs}[x]^2+\text{Abs}[-1+y]^2\right)^{3/2}}+\frac{\text{ke nC x}}{\left(\text{Abs}[x]^2+\text{Abs}[1+y]^2\right)^{3/2}},\ \frac{\text{ke nC }(-1+y)}{\left(\text{Abs}[x]^2+\text{Abs}[-1+y]^2\right)^{3/2}}+\frac{\text{ke nC }(1+y)}{\left(\text{Abs}[x]^2+\text{Abs}[1+y]^2\right)^{3/2}}\right\}$$

In[38]:= **ef2D = FullSimplify[%, Assumptions → {x ∈ Reals, y ∈ Reals}] /. repSI**

Out[38]=

$$\left\{9.\,x\left(\frac{1}{\left(x^2+(-1+y)^2\right)^{3/2}}+\frac{1}{\left(x^2+(1+y)^2\right)^{3/2}}\right),\ 9.\left(\frac{-1+y}{\left(x^2+(-1+y)^2\right)^{3/2}}+\frac{1+y}{\left(x^2+(1+y)^2\right)^{3/2}}\right)\right\}$$

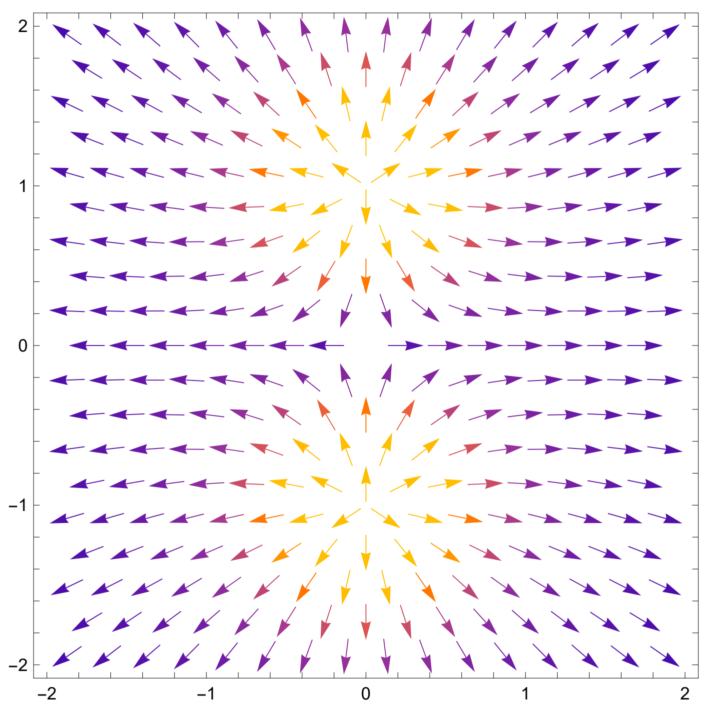
In[39]:= **(*Calculation of field at arb. point*)**
ef2D /. {x → 1, y → .45}(*as in fig.5*)

Out[39]=

{7.7014, -0.94191}

In[40]:= **VectorPlot[ef2D, {x, -2, 2}, {y, -2, 2}]**

Out[40]=



In[41]:= **figVecPlot2D == %;**

In[42]:= **Export["figVecPlot2D.jpeg", %, "JPEG"]**

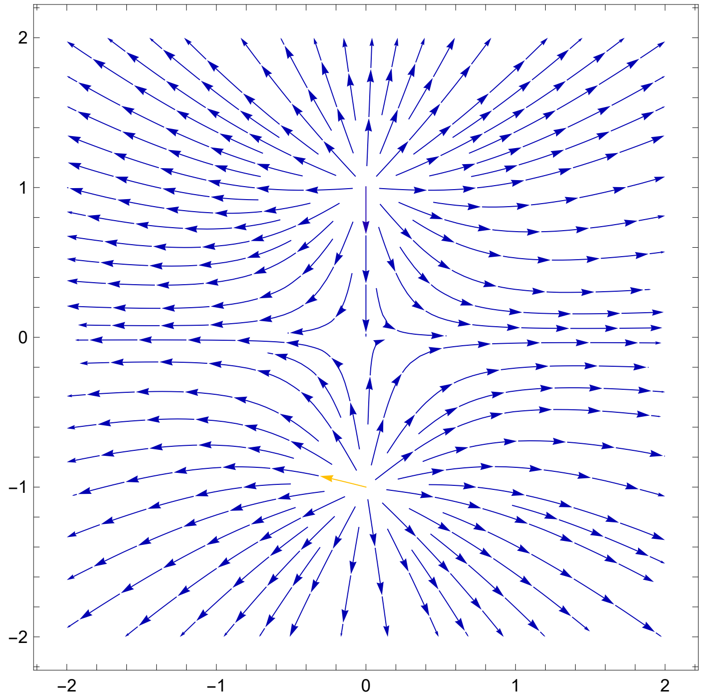
Out[42]=

figVecPlot2D.jpeg

In[43]:= **(*TRY DIFFERENT FORMATS: select the best. Print out separate figure**
(or page from notebook). Sketch lines by hand.Compare with below*)

In[44]:=
```
StreamPlot[ef2D, {x, -2, 2}, {y, -2, 2}]
```

Out[44]=



In[45]:=
```
(*COMPLETE  THE TASKS 4-6 on p.26 for a dipole. Note that VectorPlot
  in Mathematica is the closest (not identical) to quiver  in MATLAB*)
```