In[ ]:=
```
FrontEndExecute[FrontEndToken["DeleteGeneratedCells"]]
```

Problem 1

In[ ]:=
```
Clear["Global`*"]
```

In[ ]:=
```
{x, y, z} = DiagonalMatrix[{1, 1, 1}];
{x, y, z} // TableForm
```

Out[ ]//TableForm=

1	0	0
0	1	0
0	0	1

In[ ]:=
```
? Dot
```

Out[ ]=

Symbol

$a.b.c$ or Dot[a, b, c] gives products of vectors, matrices, and tensors.

In[ ]:=
```
tmp1 = Outer[dot, {"x", "y", "z"}, {"x", "y", "z"}];
tmp1 // MatrixForm
```

Out[ ]//MatrixForm=

$$\begin{pmatrix} \text{dot[x, x]} & \text{dot[x, y]} & \text{dot[x, z]} \\ \text{dot[y, x]} & \text{dot[y, y]} & \text{dot[y, z]} \\ \text{dot[z, x]} & \text{dot[z, y]} & \text{dot[z, z]} \end{pmatrix}$$

In[ ]:=
```
tmp2 = Outer[Dot, {x, y, z}, {x, y, z}, 1];
tmp2 // TableForm[#, TableHeadings → {{"x", "y", "z"}, {"x", "y", "z"}}] &
```

Out[ ]//TableForm=

	x	y	z
x	1	0	0
y	0	1	0
z	0	0	1

In[ ]:=
```
{tmp1, tmp2} // Transpose // TableForm
```

Out[ ]//TableForm=

dot[x, x]	1
dot[x, y]	0
dot[x, z]	0
dot[y, x]	0
dot[y, y]	1
dot[y, z]	0
dot[z, x]	0
dot[z, y]	0
dot[z, z]	1

In[•]:= ? Cross

Out[•]=

Symbol

Cross [a, b] gives the vector cross product of a and b.

In[•]:=
```
tmp3 = Outer[cross, {"x", "y", "z"}, {"x", "y", "z"}];
tmp3 // MatrixForm
```

Out[•]//MatrixForm=

$$\begin{pmatrix} \text{cross}[x, x] & \text{cross}[x, y] & \text{cross}[x, z] \\ \text{cross}[y, x] & \text{cross}[y, y] & \text{cross}[y, z] \\ \text{cross}[z, x] & \text{cross}[z, y] & \text{cross}[z, z] \end{pmatrix}$$

In[•]:=
```
tmp4 = Outer[Cross, {x, y, z}, {x, y, z}, 1];
tmp4 // MatrixForm[#, TableHeadings → {{"x", "y", "z"}, {"x", "y", "z"}}] &
```

Out[•]//MatrixForm=

$$\left(\begin{array}{c|ccc} & x & y & z \\ \hline x & \begin{pmatrix}0\\0\\0\end{pmatrix} & \begin{pmatrix}0\\0\\1\end{pmatrix} & \begin{pmatrix}0\\-1\\0\end{pmatrix} \\ y & \begin{pmatrix}0\\0\\-1\end{pmatrix} & \begin{pmatrix}0\\0\\0\end{pmatrix} & \begin{pmatrix}1\\0\\0\end{pmatrix} \\ z & \begin{pmatrix}0\\1\\0\end{pmatrix} & \begin{pmatrix}-1\\0\\0\end{pmatrix} & \begin{pmatrix}0\\0\\0\end{pmatrix} \end{array}\right)$$

In[•]:=
```
{tmp3, tmp4} // Transpose // TableForm
```

Out[•]//TableForm=

```
cross[x, x]    0  0   0
cross[x, y]    0  0   1
cross[x, z]    0  -1  0

cross[y, x]    0  0  -1
cross[y, y]    0  0  0
cross[y, z]    1  0  0

cross[z, x]    0   1  0
cross[z, y]    -1  0  0
cross[z, z]    0   0  0
```

In[•]:=
```
{tmp3, tmp4} // Transpose // Flatten[#, 1] & // MatrixForm[#, TableDepth → 3] &
```

Out[•]//MatrixForm=

$$\begin{pmatrix} \text{cross}[x, x] & \text{cross}[x, y] & \text{cross}[x, z] \\ \{0, 0, 0\} & \{0, 0, 1\} & \{0, -1, 0\} \\ \text{cross}[y, x] & \text{cross}[y, y] & \text{cross}[y, z] \\ \{0, 0, -1\} & \{0, 0, 0\} & \{1, 0, 0\} \\ \text{cross}[z, x] & \text{cross}[z, y] & \text{cross}[z, z] \\ \{0, 1, 0\} & \{-1, 0, 0\} & \{0, 0, 0\} \end{pmatrix}$$

Problem 2

In[ ]:=
```
Clear["Global`*"]
```

In[ ]:=
```
f1 = {x, y, 0};
f2 = {-y, x, 0};
```

In[ ]:=
```
? Div
```

Out[ ]=

Symbol

Div[{f_1, …, f_n}, {x_1, …, x_n}] gives the divergence $\partial f_1/\partial x_1 + \ldots + \partial f_n/\partial x_n$.

Div[{f_1, …, f_n}, {x_1, …, x_n}, *chart*] gives the divergence in the coordinates *chart*.

In[ ]:=
```
Div[f1, {x, y, z}]
```

Out[ ]= 2

In[ ]:=
```
Div[f2, {x, y, z}]
```

Out[ ]= 0

In[ ]:=
```
? Curl
```

Out[ ]=

Symbol

Curl[{f_1, f_2}, {x_1, x_2}] gives the curl $\partial f_2/\partial x_1 - \partial f_1/\partial x_2$.

Curl[{f_1, f_2, f_3}, {x_1, x_2, x_3}] gives the curl $(\partial f_3/\partial x_2 - \partial f_2/\partial x_3, \partial f_1/\partial x_3 - \partial f_3/\partial x_1, \partial f_2/\partial x_1 - \partial f_1/\partial x_2)$.

Curl[f, {x_1, …, x_n}] gives the curl of the $n \times n \times \ldots \times n$ array f with respect to the n-dimensional vector {x_1, …, x_n}.

Curl[f, x, *chart*] gives the curl in the coordinates *chart*.

In[ ]:=
```
Curl[f1, {x, y, z}]
```

Out[ ]= {0, 0, 0}

In[ ]:=
```
Curl[f2, {x, y, z}]
```

Out[ ]= {0, 0, 2}

In[ ]:= ? VectorPlot

Symbol

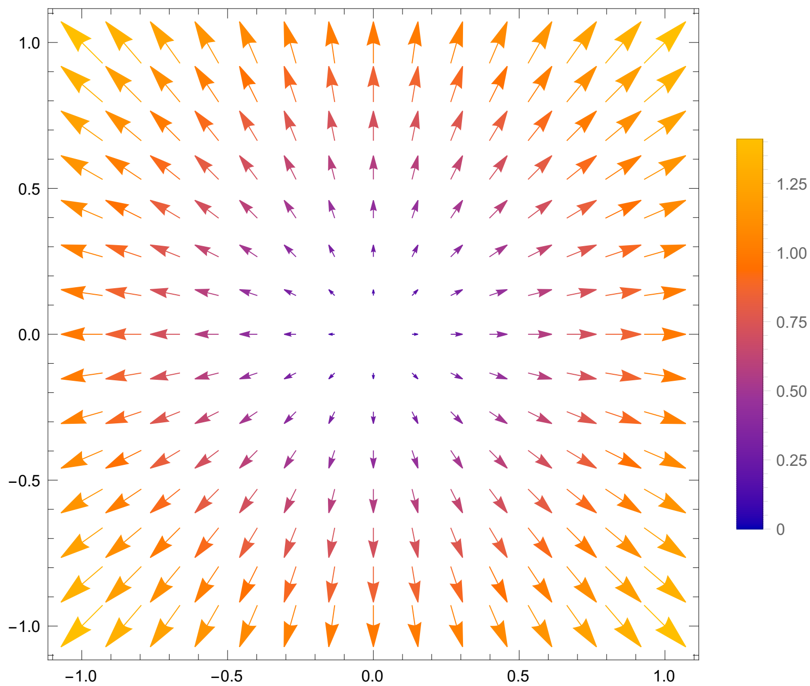
Out[ ]= VectorPlot $[\{v_x, v_y\}, \{x, x_{min}, x_{max}\}, \{y, y_{min}, y_{max}\}]$ generates a vector plot of the vector field $\{v_x, v_y\}$ as a function of x and y.

VectorPlot $[\{\{v_x, v_y\}, \{w_x, w_y\}, \ldots\}, \{x, x_{min}, x_{max}\}, \{y, y_{min}, y_{max}\}]$ plots several vector fields.

VectorPlot $[\ldots, \{x, y\} \in reg]$ takes the variables $\{x, y\}$ to be in the geometric region reg.

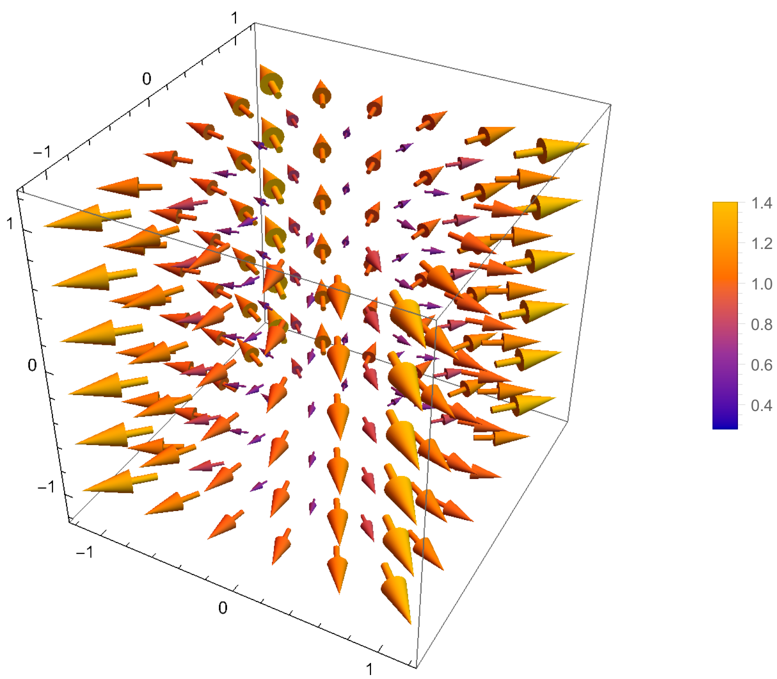
In[ ]:=
```
VectorPlot[f1[[1 ;; 2]], {x, -1, 1}, {y, -1, 1},
 PlotLegends → Automatic, VectorScale → Automatic]
```

Out[ ]=



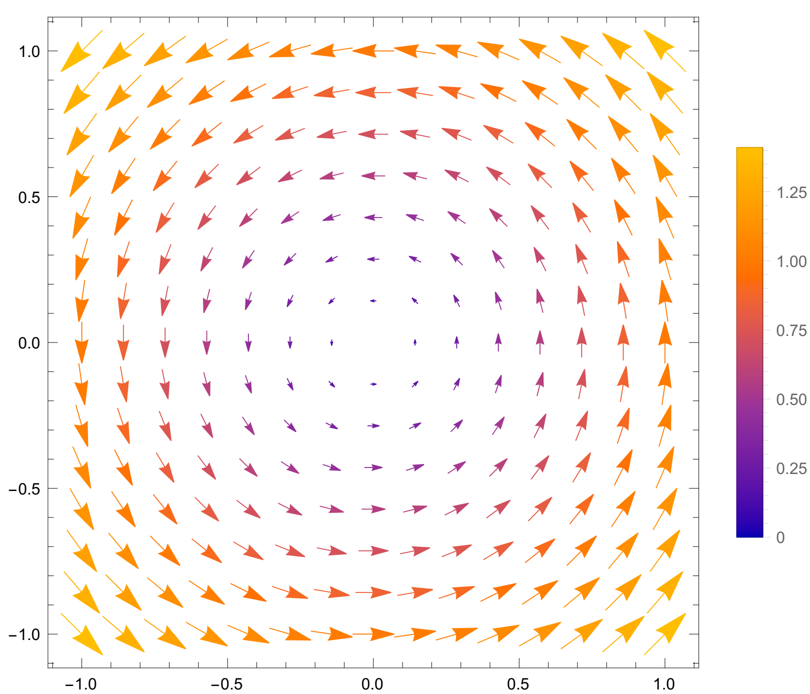
```
In[ ]:= VectorPlot3D[f1, {x, -1, 1}, {y, -1, 1}, {z, -1, 1},
         PlotLegends → Automatic, VectorScale → Automatic]
```

Out[ ]=



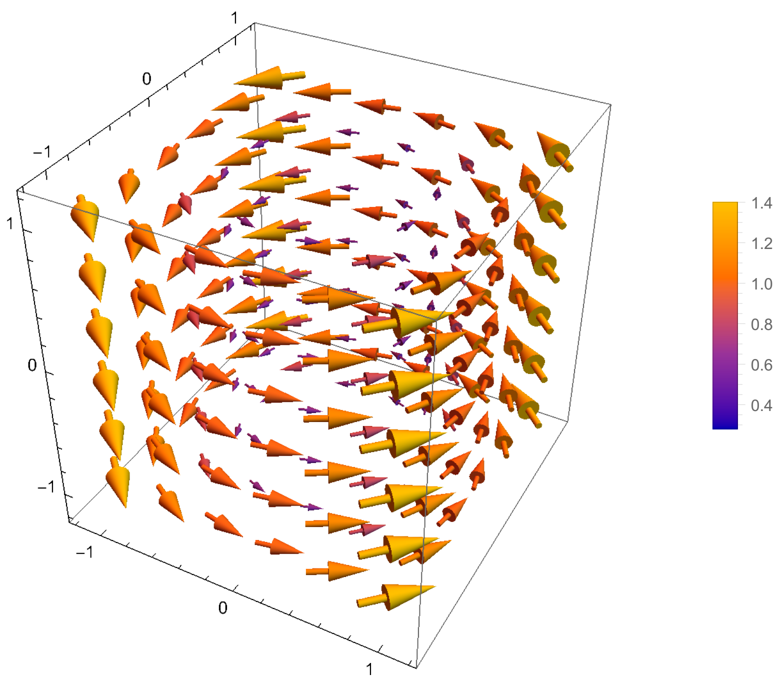

```
In[ ]:= VectorPlot[f2[[1 ;; 2]], {x, -1, 1}, {y, -1, 1},
         PlotLegends → Automatic, VectorScale → Automatic]
```

Out[ ]=



In[ ]:=
```
VectorPlot3D[f2, {x, -1, 1}, {y, -1, 1}, {z, -1, 1},
  PlotLegends → Automatic, VectorScale → Automatic]
```

Out[ ]=



Problem 3

In[ ]:=
```
Clear["Global`*"]
```

In[ ]:=
```
f1 = {r, 0, 0};
f2 = {1 / r ^ 2, 0, 0};
```

In[ ]:=
```
Div[f1, {r, θ, z}, "Cylindrical"]
```

Out[ ]= 2

In[ ]:=
```
Div[f1, {r, θ, ϕ}, "Spherical"]
```

Out[ ]= 3

In[ ]:=
```
Div[f2, {r, θ, z}, "Cylindrical"]
```

Out[ ]= $-\frac{1}{r^3}$

In[ ]:=
```
Div[f2, {r, θ, z}, "Spherical"]
```

Out[ ]= 0

In[ ]:=
```
Curl[f1, {r, θ, ϕ}, "Cylindrical"]
```

Out[ ]= {0, 0, 0}

In[ ]:= `Curl[f1, {r, θ, ϕ}, "Spherical"]`

Out[ ]= {0, 0, 0}

In[ ]:= `Curl[f2, {r, θ, ϕ}, "Cylindrical"]`

Out[ ]= {0, 0, 0}

In[ ]:= `Curl[f2, {r, θ, ϕ}, "Spherical"]`

Out[ ]= {0, 0, 0}

In[ ]:= `tx1 = {x, y, z} → CoordinateTransform["Spherical" → "Cartesian", {r, θ, ϕ}] // Thread`

Out[ ]= $\{x \to r \cos[\phi] \sin[\theta],\ y \to r \sin[\theta] \sin[\phi],\ z \to r \cos[\theta]\}$

In[ ]:= `tx2 = {r, θ, ϕ} → CoordinateTransform["Cartesian" → "Spherical", {x, y, z}] // Thread`

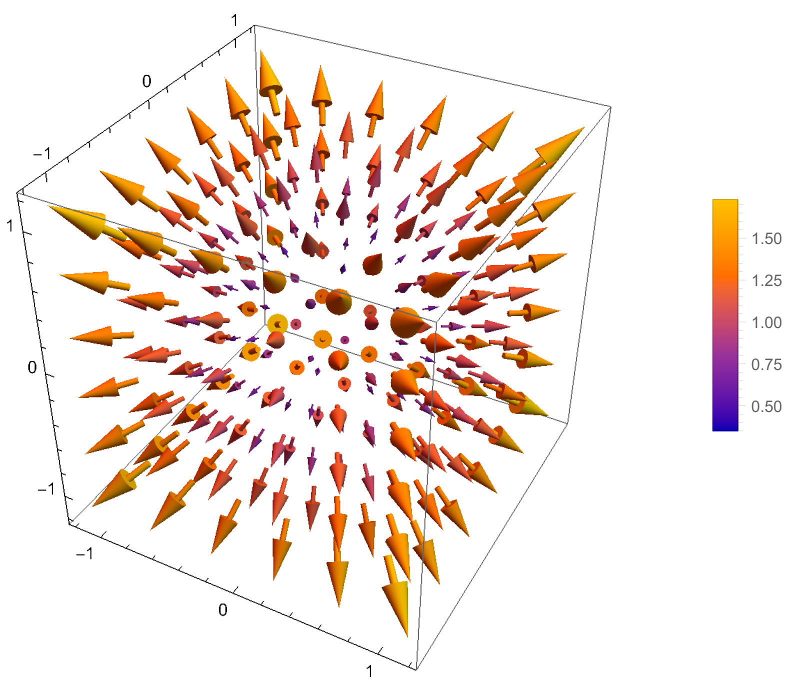
Out[ ]= $\left\{r \to \sqrt{x^2 + y^2 + z^2},\ \theta \to \text{ArcTan}\left[z, \sqrt{x^2 + y^2}\right],\ \phi \to \text{ArcTan}[x, y]\right\}$

In[ ]:= `f1 /. tx2`

Out[ ]= $\left\{\sqrt{x^2 + y^2 + z^2},\ 0,\ 0\right\}$

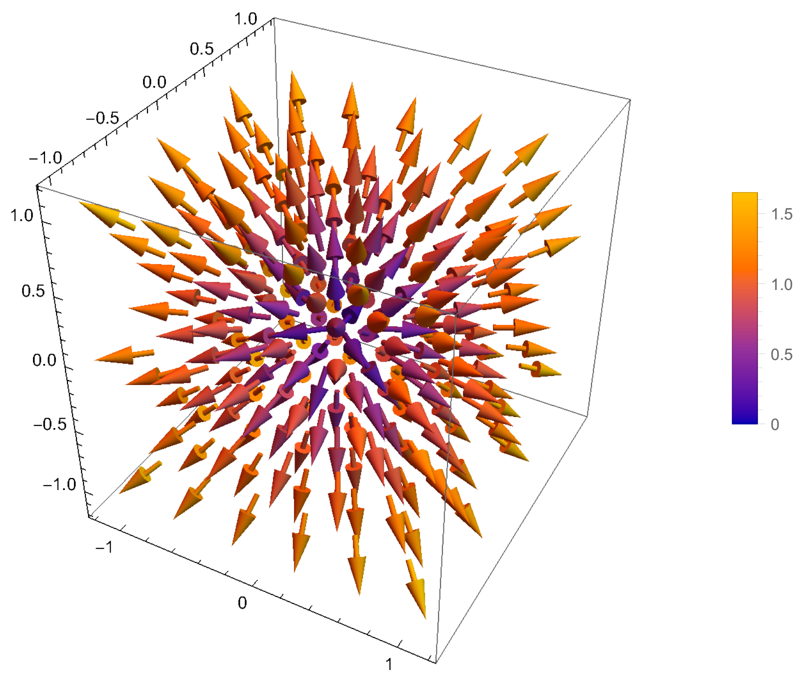
In[ ]:= `VectorPlot3D[{x, y, z} /. tx2, {x, -1, 1}, {y, -1, 1}, {z, -1, 1}, PlotLegends → Automatic, VectorScale → Automatic]`

Out[ ]=



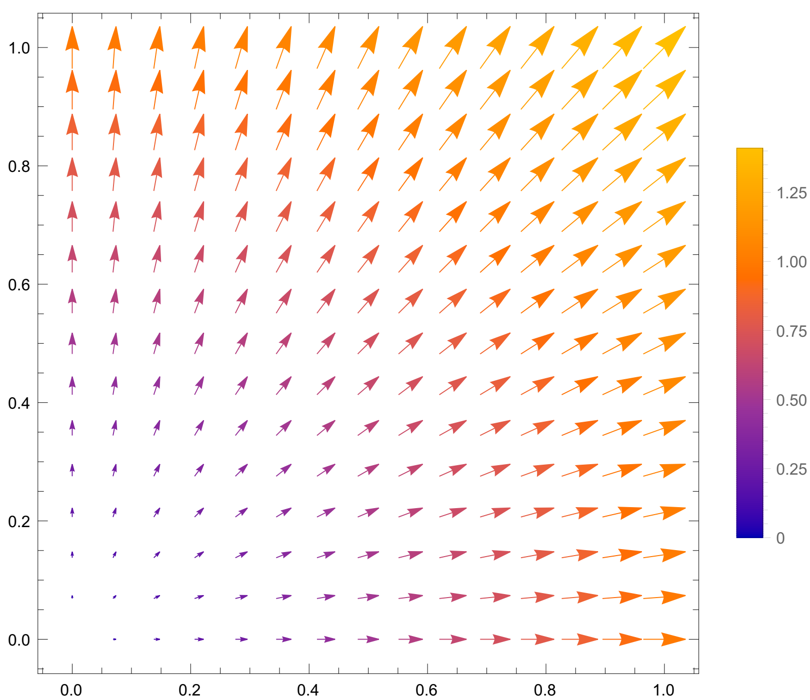
In[ ]:=
```
VectorPlot3D[{x, y, z} /. tx2, {x, -1, 1}, {y, -1, 1},
  {z, -1, 1}, PlotLegends → Automatic, VectorScale → None]
```

Out[ ]=



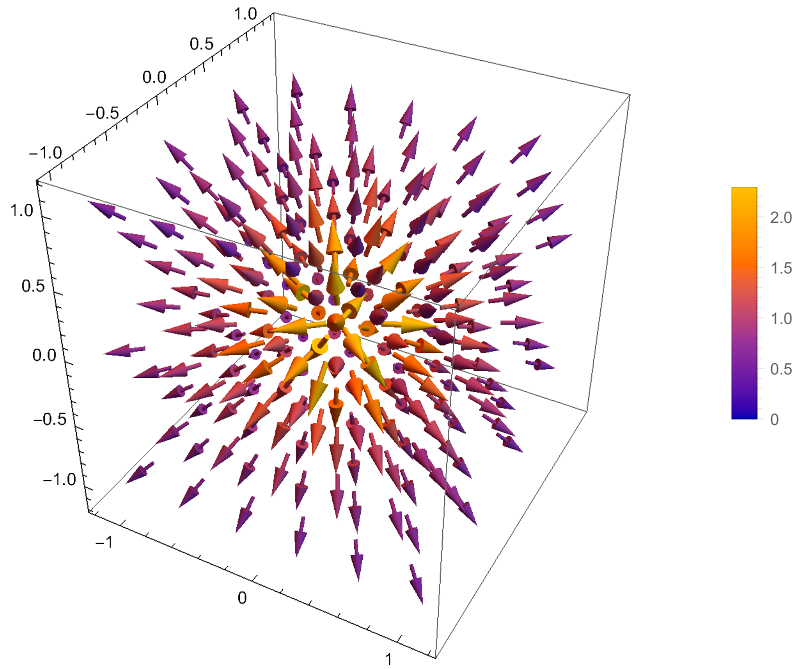
In[ ]:=
```
VectorPlot[({x, y, z} /. tx2)[[1 ;; 2]] /. {z → 0}, {x, 0, 1},
  {y, 0, 1}, PlotLegends → Automatic, VectorScale → Automatic]
```

Out[ ]=



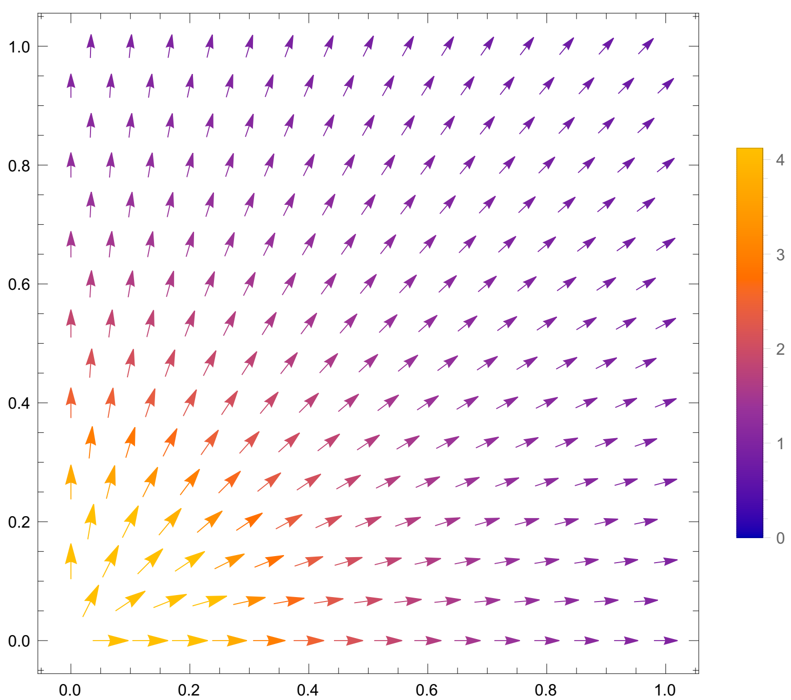
In[ ]:=
```
VectorPlot3D[1/r^2 {x, y, z} /. tx2, {x, -1, 1}, {y, -1, 1},
  {z, -1, 1}, PlotLegends → Automatic, VectorScaling → Automatic]
```

Out[ ]=



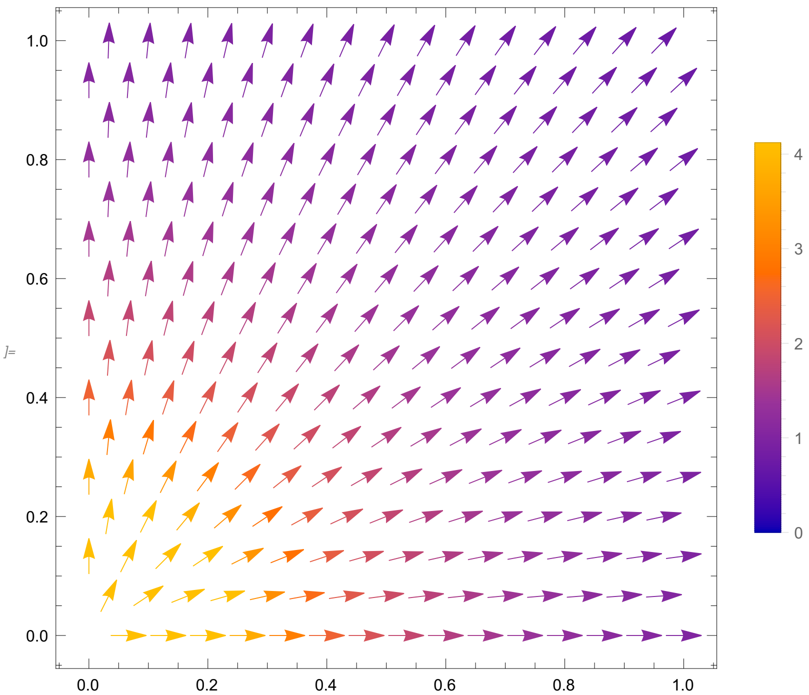
In[ ]:=
```
VectorPlot[(1/r^2 {x, y, z} /. tx2)[[1 ;; 2]] /. {z → 0}, {x, 0, 1},
  {y, 0, 1}, PlotLegends → Automatic, VectorScaling → Automatic]
```

Out[ ]=



```
In[ ]:= VectorPlot[(1 / r ^ 2 {x, y, z} /. tx2)[[1 ;; 2]] /. {z → 0},
          {x, 0, 1}, {y, 0, 1}, PlotLegends → Automatic, VectorScale → None]
```

Out[ ]=



Problem 4

```
In[ ]:= Clear["Global`*"]

In[ ]:= f1 = {0, r, 0};
        f2 = {0, 0, r Sin[θ]};

In[ ]:= Div[f1, {r, θ, z}, "Cylindrical"]

Out[ ]= 0

In[ ]:= Curl[f1, {r, θ, ϕ}, "Cylindrical"]

Out[ ]= {0, 0, 2}

In[ ]:= Div[f2, {r, θ, z}, "Spherical"]

Out[ ]= 0

In[ ]:= Curl[f2, {r, θ, ϕ}, "Spherical"]

Out[ ]= {2 Cos[θ], -2 Sin[θ], 0}
```

Problem 5

```
In[1]:= Clear["Global`*"]
```

In[2]:=
```
eqy = y == y0 + vy0 t + (1 / 2) (- g ) t ^ 2
```

Out[2]= $y == -\frac{g\, t^2}{2} + t\, vy0 + y0$

In[3]:=
```
eqx = x == x0 + vx0 t
```

Out[3]= $x == t\, vx0 + x0$

In[4]:=
```
values = {x0 → 0, y0 → h, vx0 -> v0 Cos[θ], vy0 -> v0 Sin[θ]};
```

In[5]:=
```
tRoot = Roots[eqy /. {y → 0}, t]
```

Out[5]= $t == \frac{vy0 - \sqrt{vy0^2 + 2\, g\, y0}}{g} \;||\; t == \frac{vy0 + \sqrt{vy0^2 + 2\, g\, y0}}{g}$

In[6]:=
```
(* _IF_ it  were to start and end at same height  *)
tRoot /. {y0 → 0} // PowerExpand
```

Out[6]= $t == 0 \;||\; t == \frac{2\, vy0}{g}$

In[7]:=
```
sol1 = Solve[tRoot[[2]], t][[1]] (* Take the positive root *)
```

Out[7]= $\left\{t \to \frac{vy0 + \sqrt{vy0^2 + 2\, g\, y0}}{g}\right\}$

In[8]:=
```
xSol = Solve[eqx /. sol1 , x][[1]]
```

Out[8]= $\left\{x \to \frac{vx0\, vy0 + g\, x0 + vx0\, \sqrt{vy0^2 + 2\, g\, y0}}{g}\right\}$

In[9]:=
```
xSol /. values // FullSimplify
```

Out[9]= $\left\{x \to \frac{v0\, Cos[\theta] \left(v0\, Sin[\theta] + \sqrt{2\, g\, h + v0^2\, Sin[\theta]^2}\right)}{g}\right\}$

In[10]:=
```
xSol /. values /. {h → 0} // PowerExpand // FullSimplify
```

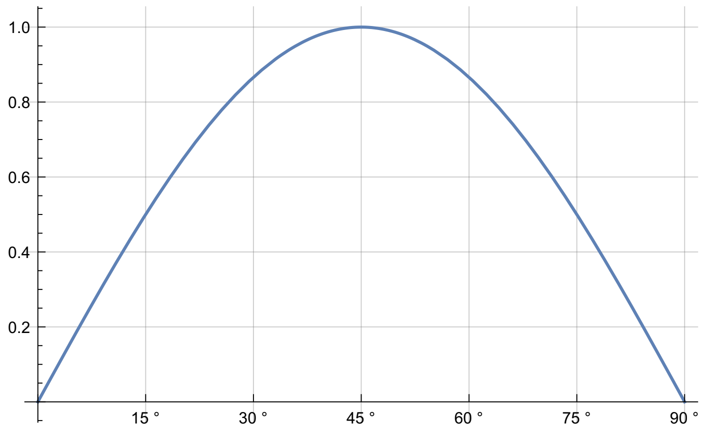
Out[10]= $\left\{x \to \frac{v0^2\, Sin[2\, \theta]}{g}\right\}$

In[11]:=
```
ticks = 15 Range[0, 6] Degree
```

Out[11]= {0, 15 °, 30 °, 45 °, 60 °, 75 °, 90 °}

In[12]:=
```
Plot[Sin[2 θ], {θ, 0, π / 2}, Ticks → {ticks , Automatic}, GridLines → {ticks , Automatic}]
```

Out[12]=



In[13]:=
```
xSol0 = Solve[eqx , x][[1]]
ySol0 = Solve[eqy , y][[1]]
```

Out[13]= $\{x \to t\ vx0 + x0\}$

Out[14]= $\left\{y \to \frac{1}{2}\left(-g\,t^2 + 2\,t\ vy0 + 2\ y0\right)\right\}$

In[15]:=
```
{x, y} /. xSol0 /. ySol0 //. values
```

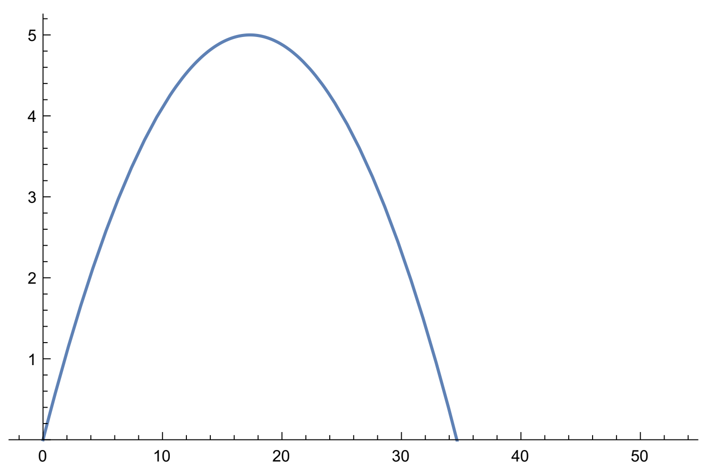
Out[15]= $\left\{t\ v0\ Cos[\theta],\ \frac{1}{2}\left(2\,h - g\,t^2 + 2\,t\ v0\ Sin[\theta]\right)\right\}$

In[16]:=
```
ff[t_, θ_, h_ : 40, v0_ : 20] = {x, y} /. xSol0 /. ySol0 //. values /. {g → +10}
SetAttributes [ff, Listable]
```

Out[16]= $\left\{t\ v0\ Cos[\theta],\ \frac{1}{2}\left(2\,h - 10\,t^2 + 2\,t\ v0\ Sin[\theta]\right)\right\}$

In[18]:=
```
ParametricPlot [ff[t, 30 Degree , 0], {t, 0, 3},
 PlotRange → {0, Automatic}, AspectRatio → 1 / GoldenRatio ]
```

Out[18]=



In[19]:=

```
list = 5 Range[17] Degree
```

Out[19]= {5 °, 10 °, 15 °, 20 °, 25 °, 30 °, 35 °, 40 °, 45 °, 50 °, 55 °, 60 °, 65 °, 70 °, 75 °, 80 °, 85 °}

In[20]:=

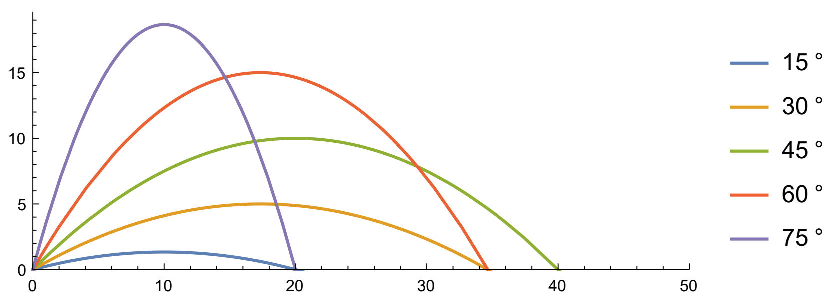
```
list = 15 Range[5] Degree
```

Out[20]= {15 °, 30 °, 45 °, 60 °, 75 °}

In[21]:=

```
ParametricPlot[ ff[t, list , 0] // Evaluate ,
 {t, 0, 10},
 PlotRange → {{0, 50}, {0, Automatic}},
 PlotLegends → list
]
```

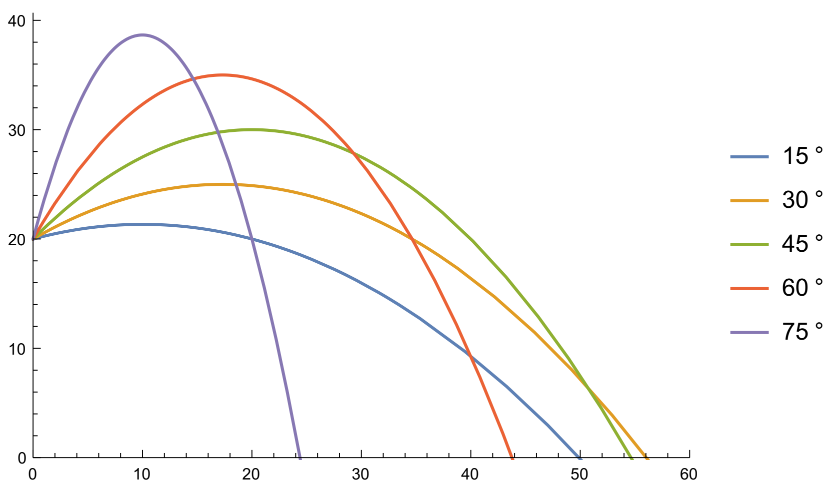
Out[21]=



In[22]:=

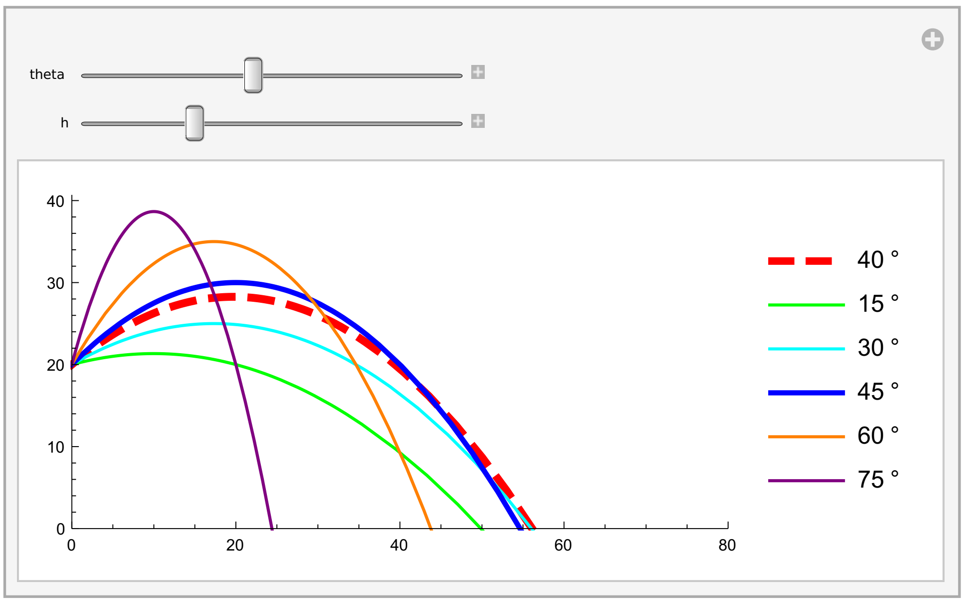
```
ParametricPlot[ ff[t, list , 20] // Evaluate ,
 {t, 0, 10},
 PlotRange → {{0, 60}, {0, Automatic}},
 PlotLegends → list
]
```

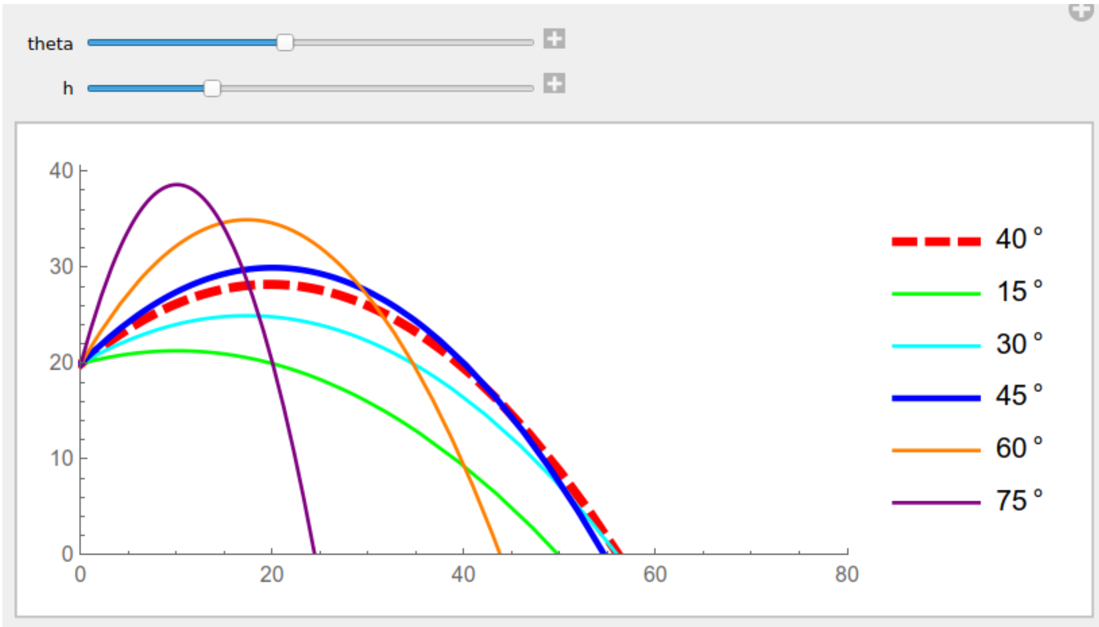
Out[22]=



In[23]:=
```
Manipulate[
 ParametricPlot[Join[{ff[t, theta Degree , h]} , ff[t, list , h]] // Evaluate ,
  {t, 0, 10},
  PlotRange → {{0, 80}, {0, Automatic}},
  PlotLegends → Join[{theta Degree} , list],
  PlotStyle → {{Thickness[0.012], Dashing[0.03], Red},
    Green , Cyan , {Thickness[0.008], Blue}, Orange , Purple , Black}
 ],
 {{theta , 40}, 0, 90, 5},
 {{h, 20}, 5, 60, 5}
]
```

Out[23]=





Max Height

In[29]:= `eqyv = 2 a (ymax - y0) == vy^2 - vy0^2`

Out[29]= $2\,a\,(-y0 + ymax) == vy^2 - vy0^2$

In[31]:= `solh = Solve[eqyv , ymax][[1]]`

Out[31]= $\left\{ymax \to \frac{vy^2 - vy0^2 + 2\,a\,y0}{2\,a}\right\}$

In[32]:= `values`

Out[32]= {x0 → 0, y0 → h, vx0 → v0 Cos[θ], vy0 → v0 Sin[θ]}

In[35]:= `solh /. values /. {a → -g, vy → 0} // Simplify`

Out[35]= $\left\{ymax \to h + \frac{v0^2\,\mathrm{Sin}[\theta]^2}{2\,g}\right\}$

Problem 6

In[◦]:= `Clear["Global`*"]`

In[◦]:= `sol = Solve[m v^2 / r == q v b, r][[1]]`

Out[◦]= $\left\{r \to \frac{m\,v}{b\,q}\right\}$

```
In[ ]:= r0 = r /. sol /. {
          m → Quantity["ProtonMass"],
          q → Quantity["ElectronCharge"],
          v → Quantity["SpeedOfLight"],
          b → Quantity[7.7, "Teslas"]}

Out[ ]= 0.12987 m_p c/(e T)

In[ ]:= radius = UnitConvert[r0, "Meters"]

Out[ ]= 0.40646 m

In[ ]:= circumference = 2 π radius

Out[ ]= 2.55386 m

In[ ]:= (* Fermilab *)
        γ = 10^3;
        γ circumference // UnitConvert[#, "Miles"] &

Out[ ]= 1.5869 mi

In[ ]:= (* LHC *)
        γ = 7 × 10^3;
        γ circumference // UnitConvert[#, "Miles"] &

Out[ ]= 11.1083 mi
```