数理解析研究所講究録
第 1865 巻 2013 年 190-203

Characteristics of a Mobile Charged Particle in Oscillating Electric Fields

Haiduke Sarafian
The Pennsylvania State University
University College
York, PA 17403
has2@psu.edu

Abstract

We consider an array of 1D and 2D rectilinear time-dependent charge distributions and evaluate their respective associated electric fields along specific directions. We then place a loose point-like charged particle in designated respective fields. The equation describing the motion of the charged particle in each case is a challenging ODE. Applying a Computer Algebra System such as *Mathematica* [1] we solve the equations numerically. Utilizing these solutions we analyze their relevant kinematic quantities. For a comprehensive understanding we simulate the motions.

keywords: Time-dependent Charge Distributions, Motion of a Charged Particle in a Time-dependent Electric Field, Numeric Solutions of ODEs, *Mathematica*.

■ Introduction and Motivation

The electrostatic interaction of two point-like charged particles is nonlinear. The motion of a loose charge in the field of a second static charge is described with a nonlinear ODE. Traditionally one attempts solving the equation analytically. Even in this "trivial" case the solution is challenging. Generalizations of this scenario based on the two-body problem aggressively are demanding. Furthermore, the equation of motion describing the movement of a charge in a time-dependent electric field generated by a second stationary charge distributions are aggressively complicated. The classic approach calls to search for analytic solutions. Although the goal is respectfully appreciative, 1) the majority of the cases the equations are not solvable 2) attempting to seek analytic solutions derails the focus and objectives of the physics of the problems, and 3) these attempts hamper the rate of scholarly production. In this article the author utilizes a Computer Algebra System, *Mathematica* and maps out a systematic approach of overcoming the aforementioned problematic issues. Beginning with a two-body point-like charged particles systematically the scenarios are generalized. For each scenario, 1) a problem is posed, 2) a solution is offered, 3) kinematics of the motion is analyzed, and finally 4) the motions are simulated for comprehensive understanding. Effectively, the entire work that is composed of text, symbolic and numeric computation, graphs and simulations are compiled in one single file. This article is composed of eight sections and a few subsections. The articles closes with a few concluding remarks.

■ Section 1

Case 1a. A two-body point-like charge-charge interaction

The problem is posed: 1) Consider a stationary point-like charged particle. Release a loose secondary charged particle in the field of the former. Assuming the charges are identical in "sign", determine the kinematics of the loose particle. 2) Repeat the scenario assuming the charge of the stationary particle is time-dependent and fluctuates with respect to time according to a sinusoidal function.

■ Solution: In order to form the equation of motion for the loose particle we apply Newton's second law, $\vec{F} = m\,\vec{a}$. The force acting on the loose particle is $\vec{F} = q\,\vec{E}$, where the field of the stationary point charge Q is $E = k\,Q_0/x^2$, in SI units

$k = 9.0 \times 10^9 \frac{N.m^2}{C^2}$. Designating the acceleration by $a \equiv \ddot{x}$ and for the sake of simplicity we set $\frac{k Q_0 q}{m} = 1$. The equation of motion becomes,

$$\ddot{x} - \frac{1}{x^2} = 0 \qquad (1)$$

To solve (1) symbolically we apply *Mathematica,*

In[1]:=
```
eqx1 = x''[t] - 1/x[t]^2;
```

In[2]:=
```
sol1 = DSolve[eqx1 == 0, x[t], t]
```

Out[2]=
$$\text{Solve}\left[\left(\frac{1}{C[1]^{3/2}}\text{Log}\left[1-\left(C[1]+\sqrt{C[1]}\sqrt{C[1]-\frac{2}{x[t]}}\right)x[t]\right]+\frac{\sqrt{C[1]-\frac{2}{x[t]}}\;x[t]}{C[1]}\right)^2 == (t+C[2])^2,\ x[t]\right]$$

Mathematica fails to solve (1) symbolically. We then try solving numerically; this requires two initial conditions. Assuming the loose particle begins at rest, $\{x(0), \dot{x}(0)\} = \{1, 0\}$ we write,

In[3]:=
```
solx11 = NDSolve[{eqx1 == 0, x[0] == 1., x'[0] == 0}, x[t], {t, 0, 2.}];
```

Table 1 is the tabulated values of the time and the position.

In[4]:=
```
tabx11 = TableForm[Table[{t, x[t] /. solx11}, {t, 0, 2.0, 0.2}],
    TableHeadings → {None, {"t,s", "x[t]"}}];
```

Although the solution of the position vs. time is given numerically, *Mathematica* allows differentiating the equation with respect to time to evaluate the corresponding velocity and acceleration. These solutions are shown in the corresponding plots.

In[5]:=
```
{velocity11, acceleration11} = {D[x[t] /. solx11, {t, 1}], D[x[t] /. solx11, {t, 2}]};
```

In[6]:=
```
plotvelocity11 = Plot[velocity11, {t, 0, 2.},
    PlotStyle → {Thick, Red}, AxesLabel → {"t,s", "v,m/s"}, GridLines → Automatic];
```

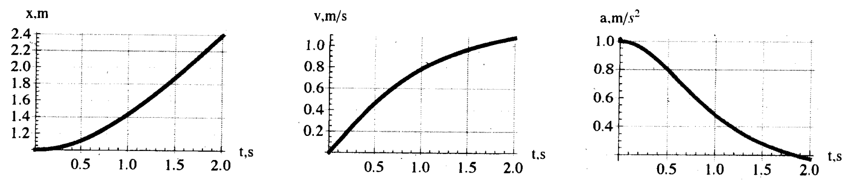
In[7]:=
```
plotacceleration11 = Plot[acceleration11, {t, 0, 2.},
    PlotStyle → {Thick, Green}, AxesLabel → {"t,s", "a,m/s²"}, GridLines → Automatic];
```

In[8]:=
```
plotx11 = Plot[x[t] /. solx11, {t, 0, 2.},
    PlotStyle → Thick, AxesLabel → {"t,s", "x,m"}, GridLines → Automatic];
```

Plots of these three quantities are shown in Fig 1.

In[9]:=
```
GraphicsGrid[{{plotx11, plotvelocity11, plotacceleration11}}, ImageSize → 500]
```

Out[9]=



■ **Case 1b.** For the oscillating stationary charge we consider $Q(t) = Q_0 \text{Cos}[2\pi f t]$. Conveniently we set the frequency

f = 5 Hz. This low frequency makes the oscillations visibly traceable. As in the previous case, we set $\frac{K Q_0 q}{m} = 1$. The equation of motion for a set if initial conditions, $\{x(0), \dot{x}(0)\} = \{0.5, 0\}$ becomes,

```
In[10]:= f = 5.; eq1f = x''[t] - Cos[2 π f t] / x[t]^2;

In[11]:= solx11f = NDSolve[{eq1f == 0, x[0] == 0.5(*1.0*), x'[0] == 0}, x[t], {t, 0, 4.}];
```

Figure 2 is the corresponding display of the associated kinematics.

```
In[12]:= plotx11f = Plot[x[t] /. solx11f, {t, 0, 4.},
           PlotStyle → Thick, AxesLabel → {"t,s", "x,m"}, GridLines → Automatic];

In[13]:= {velocity11f, acceleration11f} = {D[x[t] /. solx11f, {t, 1}], D[x[t] /. solx11f, {t, 2}]};

In[14]:= plotvelocity11f = Plot[velocity11f, {t, 0, 4.},
           PlotStyle → {Thick, Red}, AxesLabel → {"t,s", "v,m/s"}, GridLines → Automatic];

In[15]:= plotacceleration11f = Plot[acceleration11f, {t, 0, 4.},
           PlotStyle → {Thick, Green}, AxesLabel → {"t,s", "a,m/s²"}, GridLines → Automatic];

In[16]:= GraphicsGrid[{{plotx11f, plotvelocity11f, plotacceleration11f}}, ImageSize → 500]
```

Out[16]=

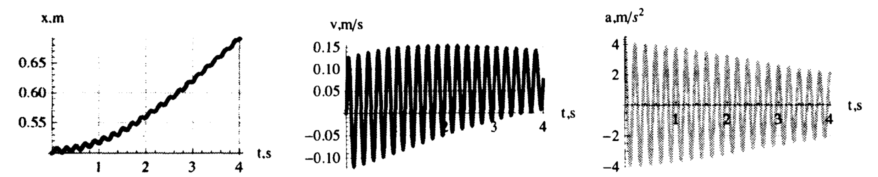
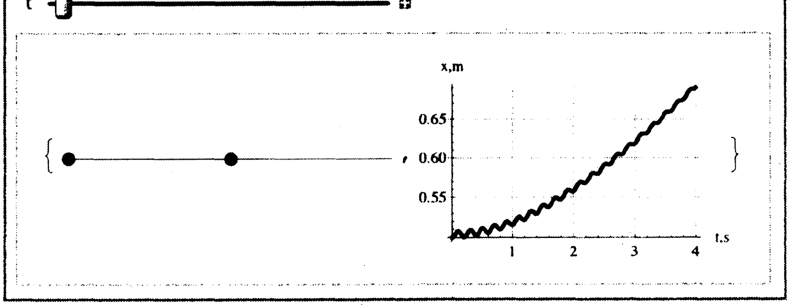


Figure 2. Display of the position, velocity and acceleration for $Q(t) = K\,Q_0\,q\,\mathrm{Cos}(2\,\pi\,f\,t)$

For a comprehensive and visual understanding we simulate the corresponding motion. Note the color of the fixed charge changes accordingly.

```
In[17]:= plotPointCharge := Graphics[{Hue[Cos[2 π f t]], Disk[{0, 0}, 0.02]}];

In[18]:= plotXaxis = Graphics[{Thin, Line[{{0, 0}, {1, 0}}]}];

In[19]:= Manipulate[{Show[{plotXaxis, plotPointCharge /. t → τ,
            Graphics[{Red, Disk[{x[t] /. solx11f[[1]] /. t → τ, 0}, 0.02]}]}, ImageSize → 200],
          plotx11f}, {{τ, 0, "t"}, 0, 4, 0.025}]
```

Out[19]=



■ **Case 2.** We consider a charged line and a point-like loose charge

The source of the static charge is a line of length ℓ. Applying basic principles [2] yields the electric field along the extension of the line, $E(x) = k\,\frac{Q_0}{x(x-\ell)}$. One may plot the $E(x)$ vs. x e.g. $\ell = 0.5\,m$.

```
In[20]:= EfieldStarightLine[x_] = 1 / (x (x - ℓ)) /. ℓ -> 0.5;

In[21]:= Plot[EfieldStarightLine[x], {x, (ℓ + 0.1) /. ℓ -> 0.5, (ℓ + 3.0) /. ℓ -> 0.5},
          PlotStyle → Thick, AxesLabel → {"x,m", "E_field, N/C"}, GridLines → Automatic];
```

The equation of motion of a loose charged particle with charge q is,

$$\ddot{x} - \frac{k\,Q_0\,q}{m}\,\frac{1}{x(x-\ell)} = 0 \tag{2}$$

Here again we set $\frac{k\,Q_0\,q}{m} = 1$. This equation numerically can be solved according to the procedure explained in the previous case. The author skipped this sets and leaves the exercise to interested reader.

For the time-dependent case we replace $K\,q\,Q_0$/m with $(K\,q\,Q_0/m)$ Cos(2 π f t) and form its equation of motion. For visual clarity we de-magnify the numeric coefficient by a factor of 0.4.

```
In[22]:= values = {ℓ -> 0.5, δ -> 0.25, (k Q0 q)/m → 1.0, f → 5.};

In[23]:= eqx2 = x''[t] - 0.4 ((k Q0 q)/m) (EfieldStarightLine[x] /. x → x[t]) Cos[2 π f t] //. values;

In[24]:= solx2 = NDSolve[{eqx2 == 0, x[0] == 0.6, x'[0] == 0}, x[t], {t, 0, 2.0}];

In[25]:= plot2 = Plot[x[t] /. solx2, {t, 0, 2.0}, PlotStyle → Thick,
          AxesLabel → {"t,s", "x,m"}, GridLines → Automatic, PlotRange → All];
```

We evaluate the velocity and the acceleration of the loose, mobile charge, and compare them to the previous case.

```
In[26]:= {velocity2f, acceleration2f} = {D[x[t] /. solx2, {t, 1}], D[x[t] /. solx2, {t, 2}]};

In[27]:= plotvelocity2f = Plot[velocity2f, {t, 0, 2.},
          PlotStyle → {Thick, Red}, AxesLabel → {"t,s", "v,m/s"}, GridLines → Automatic];

In[28]:= plotacceleration2f = Plot[acceleration2f, {t, 0.01, 2.}, PlotStyle → {Thick, Green},
          AxesLabel → {"t,s", "a,m/s^2"}, GridLines → Automatic, PlotRange → All];
```

In[29]:=
```
GraphicsGrid[{{plotx11f, plot2}, {plotvelocity11f, plotvelocity2f},
  {plotacceleration11f, plotacceleration2f}}, ImageSize → 500]
```

Out[29]=

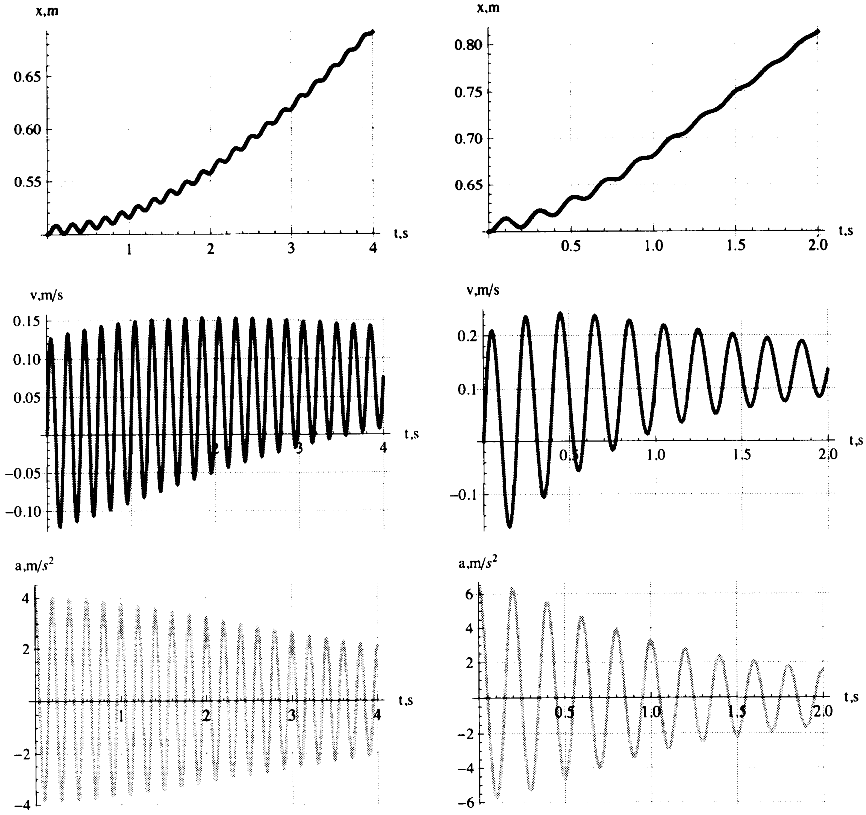


Figure 2. Display of impact of a time-independent electric field to a time-dependent field on the kinematics of the loose charge.

One notices the frequencies of the oscillations in these two cases are different. The point-like charge-charge interaction, in the first case study has the frequency of $f = 5$ Hz, the same as the frequency of the charge. On the contrary, for the second case the frequency is about $f = 3$ Hz. One interprets this as a direct impact of the shape of the charge distribution of the fixed charged source.

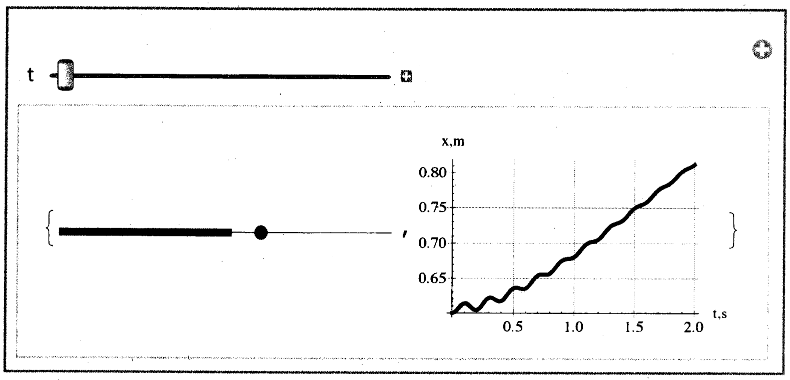
The simulation code generates the animation and also assists in visualizing the physical arrangement of the problem.

In[30]:=
```
plotChargedLine :=
  Graphics[{Thickness[0.02], Hue[Cos[2 π f t]], Line[{{0, 0}, {ℓ /. values, 0}}]}];
```

In[31]:=
```
plotXaxis = Graphics[{Thin, Line[{{0, 0}, {1, 0}}]}];
```

```
In[32]:= Manipulate[{Show[{plotXaxis, plotChargedLine /. t → τ,
      Graphics[{Red, Disk[{x[t] /. solx2[[1]] /. t → τ, 0}, 0.02]}]}, ImageSize → 200],
    plot2}, {{τ, 0, "t"}, 0, 2, 0.025}]
```

Out[32]=



■ **Case 3.** We consider two horizontal parallel charged lines. The two finite charged parallel lines of length ℓ are separated by a distance δ each with charge Q. The E-field of the given charge distribution at a point along the bisector of the lines for distances $x > \ell$ is given by, $E[x] = K\,Q_0\,[\; 2.\,/\,\ell \left(\frac{1}{\sqrt{(x-\ell)^2+\left(\frac{\delta}{2}\right)^2}} - \frac{1}{\sqrt{x^2+\left(\frac{\delta}{2}\right)^2}} \right)]$. This is a composite equation based on the field equation given in the previous case.

One may wish to plot this field. The code is given; however, due to manuscript space limitation the output is suppressed.

$$\text{In[33]:= } \mathbf{EfieldTopBottom[x\_] := 2.\,/\,\ell} \left(\frac{1}{\sqrt{(x-\ell)^2+\left(\frac{\delta}{2}\right)^2}} - \frac{1}{\sqrt{x^2+\left(\frac{\delta}{2}\right)^2}} \right) \mathbf{/.\ values}$$

```
In[34]:= Plot[EfieldTopBottom[x], {x, (ℓ + 0.1) /. values, (ℓ + 3.0) /. values},
    PlotStyle → Thick, AxesLabel → {"x,m", "E_field, N/C"}, GridLines → Automatic];
```

For time-dependent charge Q as in the previous case we consider $Q(t) = Q_0\,\mathrm{Cos}[2\,\pi\,f\,t]$. The corresponding equation of motion for a charge q becomes,

$$\text{In[35]:= } \mathbf{eqx3 = x''[t] -} \left(\frac{k\,Q_0\,q}{m}\right) \mathbf{(EfieldTopBottom[x]\ /.\ x \to x[t])\ Cos[2\,\pi\,f\,t]\ //.\ values;}$$

Assigning a set of initial conditions we solve the equation numerically.

```
In[36]:= solx3 = NDSolve[{eqx3 == 0, x[0] == 0.6, x'[0] == 0}, x[t], {t, 0, 2.0}];
```

Utilizing the solution we plot its kinematics vs. time.

```
In[37]:= plot3 = Plot[x[t] /. solx3, {t, 0, 2.0}, PlotStyle → Thick,
    AxesLabel → {"t,s", "x,m"}, GridLines → Automatic, PlotRange → All];

In[38]:= {velocity3f, acceleration3f} = {D[x[t] /. solx3, {t, 1}], D[x[t] /. solx3, {t, 2}]};

In[39]:= plotvelocity3f = Plot[velocity3f, {t, 0, 2.},
    PlotStyle → {Thick, Red}, AxesLabel → {"t,s", "v,m/s"}, GridLines → Automatic];
```

In[40]:=
```
plotacceleration3f = Plot[acceleration3f, {t, 0, 2.}, PlotStyle → {Thick, Green},
   AxesLabel → {"t,s", "a,m/s²"}, GridLines → Automatic, PlotRange → All];
```

In[41]:=
```
plotTopBottomChargedLines := Graphics[{
   Thickness[0.02], Hue[Cos[2 π f t]], Line[{{0, δ/2 /. values}, {ℓ /. values, δ/2 /. values}}],
   Thickness[0.02], Hue[Cos[2 π f t]],
   Line[{{0, -δ/2 /. values}, {ℓ /. values, -δ/2 /. values}}]}]
```

In[42]:=
```
GraphicsGrid[
 {{plotx11f, plot2, plot3}, {plotvelocity11f, plotvelocity2f, plotvelocity3f},
  {plotacceleration11f, plotacceleration2f, plotacceleration3f}}, ImageSize → 500]
```

Out[42]=

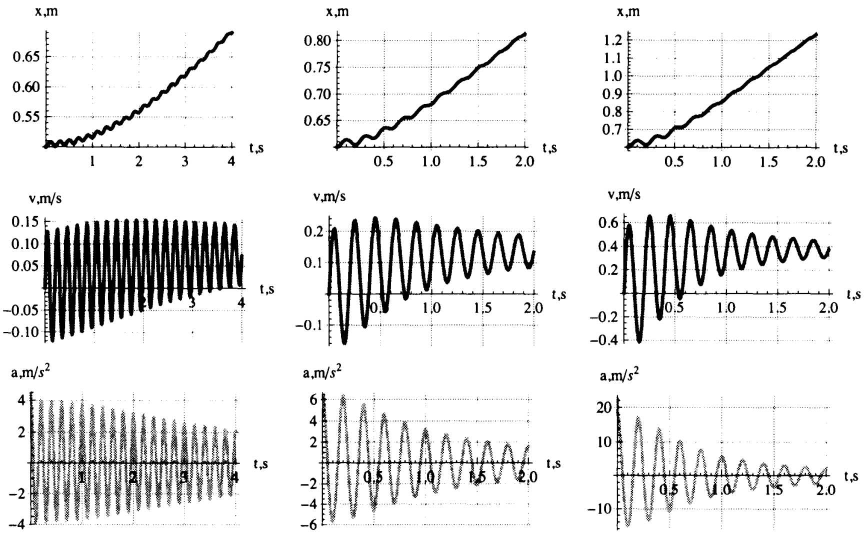


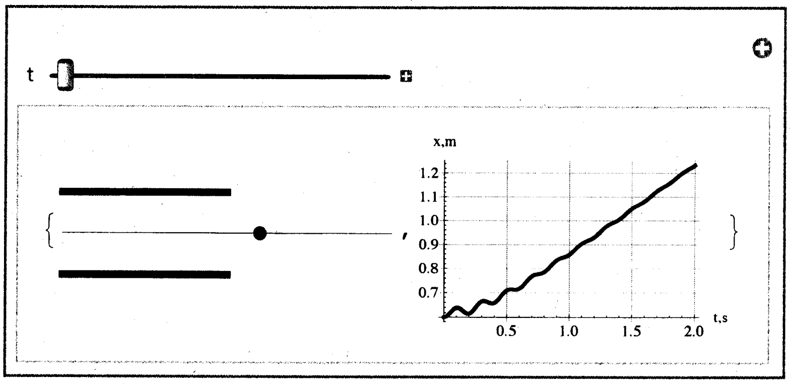
Figure 3. Display of the $\{x, v, a\}$ vs. t for case 1 through 3.

For a visual understanding we simulate the motion as well.

In[43]:=
```
plotXaxis = Graphics[{Thin, Line[{{0, 0}, {1 + 0.5, 0}}]}];
```

In[44]:=
```
Manipulate[{Show[{plotXaxis, plotTopBottomChargedLines /. t → τ,
    Graphics[{Red, Disk[{x[t] /. solx3[[1]] /. t → τ, 0}, 0.02]}]}, ImageSize → 200],
  plot3}, {{τ, 0, "t"}, 0, 2, 0.025}]
```

Out[44]=



■ **Case 4.** We consider a vertical charged line. In this scenario charge Q is distributed evenly on a vertical line of length δ. The electric field along the symmetry axis line, x. The distance away from the line is given by, $E[x] = K\,Q\,\frac{1}{x\sqrt{x^2+\left(\frac{\delta}{2}\right)^2}}$.

In[45]:=
$$\texttt{EfieldLeftVerticalLine[x\_] :=}\ \frac{1}{x\sqrt{x^2+\left(\frac{\delta}{2}\right)^2}}\ \texttt{/. values}$$

A scaled plot of the field, $\frac{1}{KQ}\,E(x)$ is shown; the output is suppressed.

In[46]:=
```
Plot[EfieldLeftVerticalLine[x], {x, (ℓ + 0.1) /. values, (ℓ + 3.0) /. values},
  PlotStyle → Thick, AxesLabel → {"x,m", "E_field, N/C"}, GridLines → Automatic];
```

The equation of motion of the corresponding field for a time-independent field is: $\ddot{x}(t) - \frac{KQq}{m}\left[\frac{1}{x\sqrt{x^2+\left(\frac{\delta}{2}\right)^2}}\right] = 0$. Setting $\frac{KQ_0 q}{m} = 1$ following the procedure given in the previous cases one may solve the equation numerically. The exercise is left to the interested of the reader. Here we solve the equation of motion for a time-dependent oscillating charge distribution. Its solution for a set of initial conditions is,

In[47]:=
$$\texttt{eqx4 = x''[t]} - \left(\frac{\mathbf{k\,Q_0\,q}}{\mathbf{m}}\right)\ \texttt{(EfieldLeftVerticalLine[x] /. x → x[t]) Cos[2 π f t] //. values;}$$

Utilizing this solution we evaluate the equation and then display its kinematics.

In[48]:=
```
solx4 = NDSolve[{eqx4 == 0, x[0] == 0.6, x'[0] == 0}, x[t], {t, 0, 2.0}];
```

In[49]:=
```
plot4 = Plot[x[t] /. solx4, {t, 0, 2.0}, PlotStyle → Thick,
  AxesLabel → {"t,s", "x,m"}, GridLines → Automatic, PlotRange → All];
```

In[50]:=
```
{velocity4f, acceleration4f} = {D[x[t] /. solx4, {t, 1}], D[x[t] /. solx4, {t, 2}]};
```

In[51]:=
```
plotvelocity4f = Plot[velocity4f, {t, 0, 2.},
  PlotStyle → {Thick, Red}, AxesLabel → {"t,s", "v,m/s"}, GridLines → Automatic];
```

In[52]:=
```
plotacceleration4f = Plot[acceleration4f, {t, 0, 2.}, PlotStyle → {Thick, Green},
   AxesLabel → {"t,s", "a,m/s²"}, GridLines → Automatic, PlotRange → All];
```

In[53]:=
```
GraphicsGrid[{{plotx11f, plot2, plot3, plot4},
   {plotvelocity11f, plotvelocity2f, plotvelocity3f, plotvelocity4f},
   {plotacceleration11f, plotacceleration2f, plotacceleration3f,
    plotacceleration4f}}, ImageSize → 500];
```

Figure 4. Display of the $\{x, v, a\}$ vs. t for case 1 through 4.

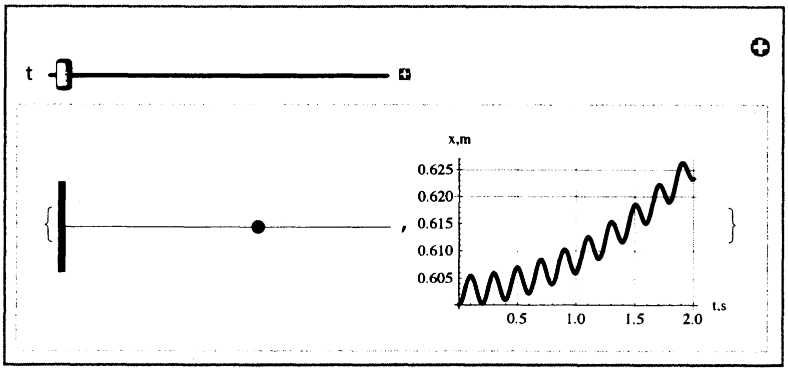
Due to manuscript space limitation the graphic output is suppressed. For a visual understanding we simulate the motion as well.

In[54]:=
```
plotVerticalChargedLine := Graphics[
   {Thickness[0.02], Hue[Cos[2 π f t]], Line[{{0, -δ/2 /. values}, {0, δ/2 /. values}}]}]
```

In[55]:=
```
plotXaxis = Graphics[{Thin, Line[{{0, 0}, {1, 0}}]}];
```

In[56]:=
```
Manipulate[{Show[{plotXaxis, plotVerticalChargedLine /. t → τ,
     Graphics[{Red, Disk[{x[t] /. solx4[[1]] /. t → τ, 0}, 0.02]}]}, ImageSize → 200],
  plot4}, {{τ, 0, "t"}, 0, 2, 0.025}]
```

Out[56]=



■ **Case 5.** We consider a horizontally displaced vertical line charge distribution. The physics of this case is similar to Case 4. In this scenario the vertical charge slides along the x-axis by a length ℓ. Replacing x with $x - \ell$ in the field equation of Case 4 yields the needed field for the case at hand.

In[57]:=
```
EfieldRightVerticalLine[x_] := 1/((x - ℓ) Sqrt[((x - ℓ)^2 + (δ/2)^2)]) /. values
```

In[58]:=
```
Plot[EfieldRightVerticalLine[x], {x, (ℓ + 0.1) /. values, (ℓ + 3.0) /. values},
   PlotStyle → Thick, AxesLabel → {"x,m", "E_field, N/C"}, GridLines → Automatic];
```

The corresponding equation of motion for a time-dependent charge distribution is,

In[59]:=
```
eqx5 = x''[t] - (k Q0 q/m) (EfieldRightVerticalLine[x] /. x → x[t]) Cos[2 π f t] //. values;
```

The numeric solution of this equation for a set of initial conditions yields the kinematics of the charge q.

In[60]:=
```
solx5 = NDSolve[{eqx5 == 0, x[0] == 0.6, x'[0] == 0}, x[t], {t, 0, 2.0}];
```

```
In[61]:= plot5 = Plot[x[t] /. solx5, {t, 0, 2.0}, PlotStyle → Thick,
           AxesLabel → {"t,s", "x,m"}, GridLines → Automatic, PlotRange → All];

In[62]:= {velocity5f, acceleration5f} = {D[x[t] /. solx5, {t, 1}], D[x[t] /. solx5, {t, 2}]};

In[63]:= plotvelocity5f = Plot[velocity5f, {t, 0, 2.}, PlotStyle → {Thick, Red},
           AxesLabel → {"t,s", "v,m/s"}, GridLines → Automatic, PlotRange → All];

In[64]:= plotacceleration5f = Plot[acceleration5f, {t, 0, 2.}, PlotStyle → {Thick, Green},
           AxesLabel → {"t,s", "a,m/s²"}, GridLines → Automatic, PlotRange → All];

In[65]:= GraphicsGrid[{{plotx11f, plot2, plot3, plot4, plot5},
           {plotvelocity11f, plotvelocity2f, plotvelocity3f, plotvelocity4f, plotvelocity5f},
           {plotacceleration11f, plotacceleration2f, plotacceleration3f,
            plotacceleration4f, plotacceleration5f}}, ImageSize → 800];
```

Figure 5. Display of the $\{x, v, a\}$ vs. t for case 1 through 5.

Due to manuscript space limitation the graphic output is suppressed.

■ **Case 6.** We consider two vertical parallel charged lines. We combine the fields of Case 4 and 5. The code to display the corresponding field is given; its output is suppressed.

```
In[66]:= Plot[EfieldLeftVerticalLine[x] + EfieldRightVerticalLine[x],
          {x, (ℓ + 0.1) /. values, (ℓ + 3.0) /. values}, PlotStyle → Thick,
          AxesLabel → {"x,m", "E_field, N/C"}, GridLines → Automatic];
```

The equation of motion, its solution and related kinematics are,

```
In[67]:= eqx45 =
           x''[t] - (k Q0 q / m) ((EfieldLeftVerticalLine[x] + EfieldRightVerticalLine[x]) /. x → x[t])
            Cos[2 π f t] //. values;

In[68]:= solx45 = NDSolve[{eqx45 == 0, x[0] == 0.6, x'[0] == 0}, x[t], {t, 0, 2.0}];

In[69]:= plot45 = Plot[x[t] /. solx45, {t, 0, 1.0}, PlotStyle → Thick,
           AxesLabel → {"t,s", "x,m"}, GridLines → Automatic, PlotRange → All];

In[70]:= {velocity45f, acceleration45f} = {D[x[t] /. solx45, {t, 1}], D[x[t] /. solx45, {t, 2}]};

In[71]:= plotvelocity45f = Plot[velocity45f, {t, 0, 2.}, PlotStyle → {Thick, Red},
           AxesLabel → {"t,s", "v,m/s"}, GridLines → Automatic, PlotRange → All];

In[72]:= plotacceleration45f = Plot[acceleration45f, {t, 0, 2.}, PlotStyle → {Thick, Green},
           AxesLabel → {"t,s", "a,m/s²"}, GridLines → Automatic, PlotRange → All];

In[73]:= GraphicsGrid[{{plotx11f, plot2, plot3, plot4, plot5, plot45},
           {plotvelocity11f, plotvelocity2f, plotvelocity3f, plotvelocity4f, plotvelocity5f,
            plotvelocity45f}, {plotacceleration11f, plotacceleration2f, plotacceleration3f,
            plotacceleration4f, plotacceleration5f, plotacceleration45f}}, ImageSize → 800];
```

Due to manuscript space limitation the graphic output is suppressed.

Figure 6. Display of the $\{x, v, a\}$ vs. t for case 1 through 5.

■ **Case 7.** We consider a horizontal one-end-closed one-end-open rectangular charged box. This scenario is generated by combining Case 3 and Case 4. Schematically this case is shown in Figure 7a.

The plot code for the electric field is given, and its output is suppressed.

```
In[74]:= Plot[EfieldTopBottom[x] + EfieldLeftVerticalLine[x],
  {x, (ℓ + 0.1) /. values, (ℓ + 3.0) /. values}, PlotStyle → Thick,
  AxesLabel → {"x,m", "E_field, N/C"}, GridLines → Automatic];

In[75]:= eqx34 = x''[t] - (k Q0 q / m)
    ((EfieldTopBottom[x] + EfieldLeftVerticalLine[x]) /. x → x[t]) Cos[2 π f t] //. values;

In[76]:= solx34 = NDSolve[{eqx34 == 0, x[0] == 0.6, x'[0] == 0}, x[t], {t, 0, 2.0}];

In[77]:= plot34 = Plot[x[t] /. solx34, {t, 0, 2.0}, PlotStyle → Thick,
  AxesLabel → {"t,s", "x,m"}, GridLines → Automatic, PlotRange → All];

In[78]:= {velocity34f, acceleration34f} = {D[x[t] /. solx34, {t, 1}], D[x[t] /. solx34, {t, 2}]};

In[79]:= plotvelocity34f = Plot[velocity34f, {t, 0, 2.}, PlotStyle → {Thick, Red},
  AxesLabel → {"t,s", "v,m/s"}, GridLines → Automatic, PlotRange → All];

In[80]:= plotacceleration34f = Plot[acceleration34f, {t, 0, 2.}, PlotStyle → {Thick, Green},
  AxesLabel → {"t,s", "a,m/s^2"}, GridLines → Automatic, PlotRange → All];

In[81]:= GraphicsGrid[{{plotx11f, plot2, plot3, plot4, plot5, plot45, plot34},
  {plotvelocity11f, plotvelocity2f, plotvelocity3f,
   plotvelocity4f, plotvelocity5f, plotvelocity45f, plotvelocity34f},
  {plotacceleration11f, plotacceleration2f, plotacceleration3f, plotacceleration4f,
   plotacceleration5f, plotacceleration45f, plotacceleration34f}}, ImageSize → 800];
```

Due to manuscript space limitation the graphic output is suppressed.

Figure 7. Display of the $\{x, v, a\}$ vs. t for case 1 through 6.

For a visual understanding we simulate the motion as well.

```
In[82]:= plotRotatedCup := Graphics[{Thickness[0.02],
   Hue[Cos[2 π f t]], Line[{{0, -δ/2 /. values}, {ℓ /. values, -δ/2 /. values}}],
   Thickness[0.02], Hue[Cos[2 π f t]],
   Line[{{0, δ/2 /. values}, {ℓ /. values, δ/2 /. values}}],
   Thickness[0.02], Hue[Cos[2 π f t]], Line[{{0, -δ/2 /. values}, {0, δ/2 /. values}}]}]

In[83]:= plotXaxis = Graphics[{Thin, Line[{{0, 0}, {1 + 0.5, 0}}]}];
```

```
In[84]:= Manipulate[{Show[{plotXaxis, plotRotatedCup /. t → τ,
      Graphics[{Red, Disk[{x[t] /. solx34[[1]] /. t → τ, 0}, 0.02]}]}, ImageSize → 200],
    plot34}, {{τ, 0, "t"}, 0, 1.2, 0.025}]
```

Out[84]=

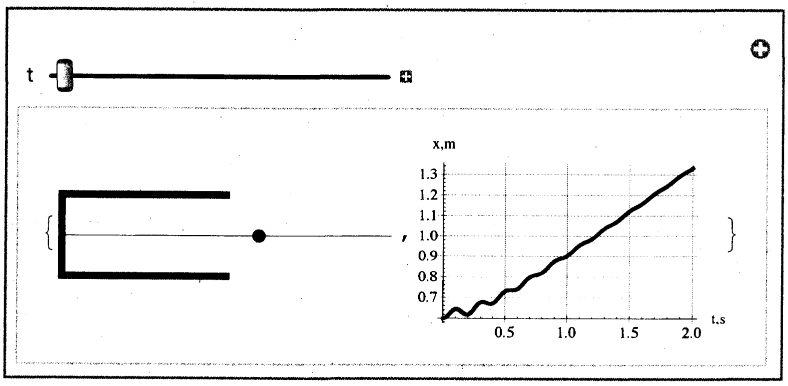


Figure 7a. Schematic of the one-end-closed one-end-open rectangular charge distribution (left graph), and position vs. time (right graph).

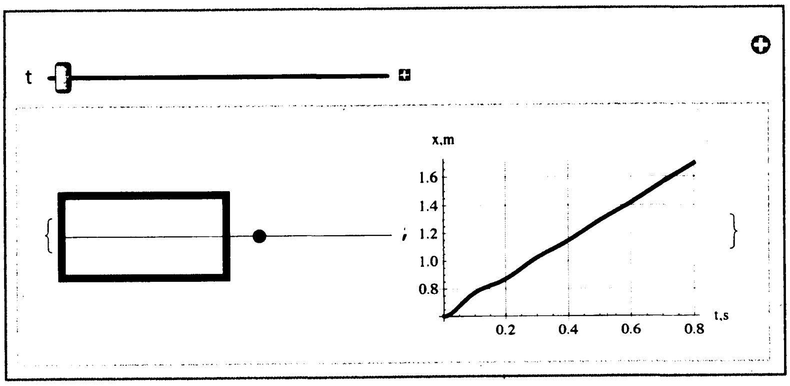
■ **Case 8.** We consider a charged rectangular closed box. By combining the configurations of Case 3, 4 and 5 we arrive at the field of a rectangular charged distribution. As in the previous scenarios the relevant associated information yields,

```
In[85]:= Plot[EfieldTopBottom[x] + EfieldLeftVerticalLine[x] + EfieldRightVerticalLine[x],
    {x, (ℓ + 0.1) /. values, (ℓ + 3.0) /. values}, PlotStyle → Thick,
    AxesLabel → {"x,m", "E_field, N/C"}, GridLines → Automatic];

In[86]:= eqx345 = x''[t] - (k Q0 q / m)
      ((EfieldTopBottom[x] + EfieldLeftVerticalLine[x] + EfieldRightVerticalLine[x]) /.
        x → x[t]) Cos[2 π f t] //. values;

In[87]:= solx345 = NDSolve[{eqx345 == 0, x[0] == 0.6, x'[0] == 0}, x[t], {t, 0, 2.0}];

In[88]:= plot345 = Plot[x[t] /. solx345, {t, 0, 0.8}, PlotStyle → Thick,
     AxesLabel → {"t,s", "x,m"}, GridLines → Automatic, PlotRange → All];

In[89]:= {velocity345f, acceleration345f} = {D[x[t] /. solx345, {t, 1}], D[x[t] /. solx345, {t, 2}]};

In[90]:= plotvelocity345f = Plot[velocity345f, {t, 0, 2.}, PlotStyle → {Thick, Red},
     AxesLabel → {"t,s", "v,m/s"}, GridLines → Automatic, PlotRange → All];

In[91]:= plotacceleration345f = Plot[acceleration345f, {t, 0, 2.}, PlotStyle → {Thick, Green},
     AxesLabel → {"t,s", "a,m/s^2"}, GridLines → Automatic, PlotRange → All];

In[92]:= GraphicsGrid[{{plotx11f, plot2, plot3, plot4, plot5, plot45, plot34, plot345},
    {plotvelocity11f, plotvelocity2f, plotvelocity3f, plotvelocity4f,
     plotvelocity5f, plotvelocity45f, plotvelocity34f, plotvelocity345f},
    {plotacceleration11f, plotacceleration2f, plotacceleration3f,
     plotacceleration4f, plotacceleration5f, plotacceleration45f,
     plotacceleration34f, plotacceleration345f}}, ImageSize → 800];
```

Due to manuscript space limitation the graphic output is suppressed. We provide the motional simulation code,

```
In[93]:= plotRectangularBox := Graphics[{Thickness[0.02],
     Hue[Cos[2 π f t]], Line[{{0, -δ/2 /. values}, {ℓ /. values, -δ/2 /. values}}],
     Thickness[0.02], Hue[Cos[2 π f t]],
     Line[{{0, δ/2 /. values}, {ℓ /. values, δ/2 /. values}}],
     Thickness[0.02], Hue[Cos[2 π f t]], Line[{{0, -δ/2 /. values}, {0, δ/2 /. values}}],
     Thickness[0.02], Hue[Cos[2 π f t]],
     Line[{{ℓ /. values, -δ/2 /. values}, {ℓ /. values, δ/2 /. values}}]}]
  plotXaxis = Graphics[{Thin, Line[{{0, 0}, {1, 0}}]}];
  Manipulate[{Show[{plotXaxis, plotRectangularBox /. t → τ,
      Graphics[{Red, Disk[{x[t] /. solx345[[1]] /. t → τ, 0}, 0.02]}]}, ImageSize → 200],
    plot345}, {{τ, 0, "t"}, 0, 0.28, 0.025}]
```

Out[95]=



■ Summary and Conclusions

It is the objective of this article to demonstrate by utilizing a Computer Algebra System (CAS), particularly *Mathematica* one may deviate from the traditional route of solving problems. The ultimate objective of a physics research project is the output of the analysis and *Mathematica* provides one such innovative approach. The traditional approach to solve a mathematical-physics problem in most scenarios encounters solving complicated equations analytically. One devotes considerable efforts doing so and fails in most cases. This derails the focus on the objectives. The author believes CAS and particularly *Mathematica* is an alternative effective approach. The examples shown in this article demonstrate how effectively one can focus on the objectives of the proposed problems and conveniently without distraction achieve the set goals. The examples are chosen from electromagnetism and the proposed approach easily may be applied to other fields of interest. Also it is worthwhile pointing out that the entire manuscript including text, symbolic and numeric computations, tables, and graphs are embodied in one single file. This by itself is a tremendous advantage assisting to avoid compiling multiple individual files.

■ Acknowledgement

The author is grateful to the chair of the organizer of the Research Institute for Mathematical Sciences Symposium, Professor Yasuyuki Nakamura-san for inviting him to present this article. He is also thankful for the financial support of the RIMS of Kyoto University, Japan. The papers presented at the symposium were informative and stimulating. The author is ecstatic for having the opportunity to sit at the desk of one of his idols, Professor Hideki Yukawa-sama Nobel prize Laurent in

theoretical physics, at the Center for Theoretical Physics/Kyoto University on August 22, 2012.

▪ References

[1] Mathematica V 8.04, Wolfram Research Inc.
[2] Halliday, Resnickand Walker, "Fundamentals of Physics", 8th edition, John Wiley and Sons, Inc. 2008.
Reitz and Milford, "Foundations of Electromagnetic Theory", Addison-Wesley Publishing Company, Inc. 1960
Jackson, "Classical Electrodynamics", 2nd edition, John Wiley and Sons, Inc. 1975.