

Proton-Proton Elastic Collision with Coulomb Repulsion

(Force is assumed to be $c \frac{\hat{r}}{r^2}$)

```
(Local) In[93]:=
Clear["Global`*"];
```

■ Various constants (masses assumed equal):

```
(Local) In[94]:=
val = {c # 0.1, m1 # 1, m2 # 1};
```

■ Function for distance between the protons:

```
(Local) In[95]:=
r[t_] = Sqrt[(x1[t] - x2[t])^2 + (y1[t] - y2[t])^2];
```

■ Equations of motion in 2 dimensions ($m \vec{a} = c \frac{\hat{r}}{r^2}$):

```
(Local) In[96]:=
eq1 = {
  m1 x1''[t] # c (x1[t] - x2[t]) / r[t]^3,
  m1 y1''[t] # c (y1[t] - y2[t]) / r[t]^3,
  m2 x2''[t] # c (x2[t] - x1[t]) / r[t]^3,
  m2 y2''[t] # c (y2[t] - y1[t]) / r[t]^3};
```

■ Initial conditions:

$y1[0]$ is impact parameter - increase for more glancing collisions

$x2[0] = +10$, $x2'[0] = -1$ for CM collision

```
(Local) In[97]:=
init = {x1[0] # -10, x1'[0] # 1,
  y1[0] # -0.1, y1'[0] # 0,
  x2[0] # 10, x2'[0] # -1,
  y2[0] # 0, y2'[0] # 0};
```

■ Solve for trajectories numerically: get coordinates as function of t

```
(Local) In[98]:=
  eq2 = Join[eq1, init] /. val;
```

```
(Local) In[99]:=
  sol = NDSolve[eq2, {x1, y1, x2, y2}, {t, 0, 20}][[1]];
```

■ Define a plot of the proton positions as a function of t and tabulate it for ten t values from 0 to 20

```
(Local) In[100]:=
  Clear[coord1plot];
  coord1plot[t_] :=
    ListPlot[{{x1[t], y1[t]}} /. sol, PlotStyle &#224; {PointSize[0.03], RGBColor[0, 0, 1]},
      GridLines &#224; Automatic, Frame &#224; True, PlotRange &#224; {{-10, 10}, {-5, 5}}]
```

```
(Local) In[102]:=
  Clear[coord2plot];
  coord2plot[t_] :=
    ListPlot[{{x2[t], y2[t]}} /. sol, PlotStyle &#224; {PointSize[0.03], RGBColor[1, 0, 0]},
      GridLines &#224; Automatic, Frame &#224; True, PlotRange &#224; {{-10, 10}, {-5, 5}}]
```

```
(Local) In[104]:=
  plotarray = Table[{coord1plot[tp], coord2plot[tp]}, {tp, 0, 20, 2}];
```

■ Combine the ten plots into one plot

```
(Local) In[105]:=
  Show[plotarray];
```

