Streuung am Zentralpotential, Betrachtung in der x-y-Ebene

Längen in fm
Geschwindigkeiten in Vielfachen von c
Energien sind durch m c^2 geteilt, also Masse=1
Klassische Hamiltonsche Bewegungsgleichungen

Definitionen

Potentialparameter, Plotoptionen

In[∘]:=
```
a = 1.5 × 10 ^ (-3);
M = 1.0;
Rand = 10.`;
Stossparameter = 1.0;
PlotRand = Rand + 4.`;
```

1. Nur Coulomb-Potential

In[∘]:=

$$V[x\_, y\_, z\_] = \frac{a}{\left(x^2 + y^2 + z^2\right) \wedge (1/2)};$$

```
Plot[V[x, 0, 0], {x, 1, 10}]
```

In[∘]:=
```
dVdx[x_, y_, z_] = D[V[x, y, z], x]
```

In[∘]:=
```
dVdy[x_, y_, z_] = D[V[x, y, z], y]
```

Dynamik

Differentialgleichungen

In[•]:=
```
EOM = {X′[t] == PX[t] / M, Y′[t] == PY[t] / M,
   PX′[t] == -dVdx[X[t], Y[t], 0], PY′[t] == -dVdy[X[t], Y[t], 0]}
```

Anfangsbedingungen:

In[•]:=
```
AB = {X[0] == -Rand, Y[0] == Stossparameter, PX[0] == 0.1, PY[0] == 0}
```

Alle Gleichungen

In[•]:=
```
Equations = Join[EOM, AB]
```

DGL lösen

In[•]:=
```
Solution = NDSolve[Equations, {X, Y, PX, PY}, {t, 0, 300}, MaxSteps → Infinity];
```

Bewegung darstellen: Machen Sie das für einen anderen Stoßparameter.

In[•]:=
```
Plot0 = ParametricPlot[{X[t], Y[t]} /.Solution⟦1⟧, {t, 0.1`, 299.`},
   PlotRange → {{-PlotRand, PlotRand}, {-PlotRand, PlotRand}},
   BaseStyle → {FontFamily → "Helvetica", FontSize → 18}]
```

In[•]:=
```
PlotStossparameter = Table[Plot0, {i, 1, 20}];
For[i = 1, i < 21, i++,
  Stossparameter = 0.25 * i;
  AB = {X[0] == -Rand, Y[0] == Stossparameter, PX[0] == 0.1, PY[0] == 0};
  Equations = Join[EOM, AB];
  Solution = NDSolve[Equations, {X, Y, PX, PY}, {t, 0, 300}, MaxSteps → Infinity];
  PlotStossparameter[[i]] = ParametricPlot[{X[t], Y[t]} /.Solution⟦1⟧,
    {t, 0.1`, 299.`}, PlotRange → {{-Rand, Rand}, {-0.1, 6.0}},
    BaseStyle → {FontFamily → "Helvetica", FontSize → 18},
    PlotStyle → {Hue[1 - 0.05 * i]}]];
```

In[•]:=
```
S1 = Show[PlotStossparameter]
```

2. Coulomb-Potential + fiktive starke Abstoßung

In[∘]:=

```
V[x_, y_, z_] = a/((x^2 + y^2 + z^2)^(1/2)) + 1.0/((x^2 + y^2 + z^2)^(12));
Plot[V[x, 0, 0], {x, 1, 10}]
```

In[∘]:=

```
dVdx[x_, y_, z_] = D[V[x, y, z], x]
```

In[∘]:=

```
dVdy[x_, y_, z_] = D[V[x, y, z], y]
```

Dynamik

Differentialgleichungen

In[∘]:=

```
EOM = {X′[t] == PX[t] / M, Y′[t] == PY[t] / M,
   PX′[t] == -dVdx[X[t], Y[t], 0], PY′[t] == -dVdy[X[t], Y[t], 0]}
```

Anfangsbedingungen:

In[∘]:=

```
AB = {X[0] == -Rand, Y[0] == Stossparameter, PX[0] == 0.1, PY[0] == 0}
```

Alle Gleichungen

In[∘]:=

```
Equations = Join[EOM, AB]
```

DGL lösen

In[∘]:=

```
Solution = NDSolve[Equations, {X, Y, PX, PY}, {t, 0, 300}, MaxSteps → Infinity];
```

Bewegung darstellen: Nehmen Sie mal einen kleineren Stoßparameter.

In[∘]:=

```
Plot0 = ParametricPlot[{X[t], Y[t]} /.Solution⟦1⟧, {t, 0.1`, 299.`},
   PlotRange → {{-PlotRand, PlotRand}, {-PlotRand, PlotRand}},
   BaseStyle → {FontFamily → "Helvetica", FontSize → 18}]
```

In[•]:=
```
PlotStossparameter = Table[Plot0, {i, 1, 20}];
For[i = 1, i < 21, i++,
  Stossparameter = 0.25 * i;
  AB = {X[0] == -Rand, Y[0] == Stossparameter, PX[0] == 0.1, PY[0] == 0};
  Equations = Join[EOM, AB];
  Solution = NDSolve[Equations, {X, Y, PX, PY}, {t, 0, 300}, MaxSteps → Infinity];
  PlotStossparameter[[i]] = ParametricPlot[{X[t], Y[t]} /.Solution⟦1⟧,
    {t, 0.1`, 299.`}, PlotRange → {{-Rand, Rand}, {-0.1, 6.0}},
    BaseStyle → {FontFamily → "Helvetica", FontSize → 18},
    PlotStyle → {Hue[1 - 0.05 * i]}]];
```

In[•]:=
```
S2 = Show[PlotStossparameter]
```

Jetzt gibt es auch Rückstreuung! Wie würden Sie Rückstreuung definieren?

3. Vergleich

In[•]:=
```
Show[S1, S2]
```

Man kann jetzt die Ausdehnung des streuenden Objektes abschätzen über den kleinsten Stoßparameter, bei dem beide Streuungen praktisch das gleiche Ergebnis geben. In dem Beispiel etwa ... fm.