

Dokumentation zum Mathematica[®] Applet Teilchen im Zentralpotential

Anatoli Ulmer, Michael Wünscher & Sebastian Eiser

20. Februar 2014



Institut für Theoretische Physik
Technische Universität Berlin

Zusammenfassung

Diese Dokumentation beschreibt die Bedienung und Funktion eines Mathematica¹-Applets, mit dem die Zweikörperbewegung in Relativkoordinaten visualisiert wird. Besonderes Augenmerk liegt dabei auf der Allgemeinheit des Potentials. Das bekannte Keplerproblem (Planetenbewegung) mit dem Gravitationspotential $-k\frac{1}{r}$ ist nur ein Spezialfall, der behandelt werden kann.

Das Mathematica-Applet ist eine Portierung des Java-Applets *Zweikörperbewegung* von Michael Wünscher & Sebastian Eiser und wurde wegen Kompatibilitätsproblemen unterschiedlicher JVM (Java Virtual Machine) im Rahmen des *G-RISC*-Projekts (German-Russian Interdisciplinary Science Center) erstellt.

¹® Wolfram Research

Inhaltsverzeichnis

1	Einleitung	4
2	Bedienung des Applets	6
2.1	Benutzeroberfläche	6
2.2	Tipps für Vorführung	6
3	Programm	11
3.1	Programmstruktur	11
4	Theorie (kurz)	14
4.1	Beschreibung mittels Differentialgleichung	14
	Literatur	15

Impressum

Institut für Theoretische Physik
Hardenbergstr. 36, Sekr. EW7-1
10623 Berlin

e-Module zur Veranschaulichung der Theoretischen Physik
Projektleitung: *Dr. Philipp Hövel* & *Prof. Dr. Ekehard Schöll*, *PhD*
Kontakt: owl@itp.physik.tu-berlin.de

1 Einleitung

Das Mathematica-Applet dient zur Veranschaulichung der Bewegung von zwei Körpern, die miteinander wechselwirken². Einer der beiden Körper befindet sich starr im Zentrum des Applets, was durch die Verwendung von Relativkoordinaten möglich ist [2]. Die *.cdf-Datei lässt sich mit *Mathematica* (Version 6.0 und aufwärts) oder dem kostenlosen *Wolfram CDF Player* (<http://www.wolfram.com/cdf-player/>) ausführen.

Was dieses Applet kann:

- Numerisch das Zweikörperproblem für nahezu beliebige Potentiale lösen
- Bahnkurven von Massepunkten zeichnen
- Bahnkurven speichern
- Anfangsbedingungen sind variabel
- Effektive Potentiale zeichnen und die aktuellen Position der Masse anzeigen
- Den Harmonischen Oszillator (und weitere vordefinierte Potentiale) in 2 Dimensionen visualisieren

Was das Applet nicht kann:

- Dreidimensionale Darstellungen
- beliebig hohe Energien als Anfangsbedingungen akzeptieren
- die numerische Integration und Darstellung erfordert viel Rechenkapazität, sodass die Darstellung unter Umständen nicht Ruckelfrei erfolgt

Der bekannteste Fall des *Zweikörper-Problems* (und auch die Voreinstellung beim Start des Applets) ist die Planetenbewegung, auch Keplerproblem genannt. Bei der Planetenbewegung wirkt die Gravitationskraft, die mit $\frac{1}{r^2}$ (r Abstand zum Kraftursprung) abfällt, also ein $-\frac{1}{r}$ Potential aufweist.

Die Besonderheit dieses Applets besteht in der Möglichkeit, dieses Potential (also auch die Wechselwirkungskraft) mit einer Formel nahezu beliebig zu verändern. Damit können nun verschiedene physikalische oder auch hypothetische Potentiale simuliert werden. Beispiele dafür *Harmonischer Oszillator* oder das *Coulomb* Potential (nur klassisch). Aber auch Mischformen von verschiedenen Potentialen (z.B. $V(r) = -r^{-1} + r^2$), die von theoretischem Interesse sind, können visualisiert und betrachtet werden. Man kann auf diese Weise einfach sehen, dass z.B. die Bahn eines Planeten nicht mehr elliptisch ist, sobald kein reines Gravitationspotential vorliegt (siehe Abbildung 1).

Das zentralsymmetrische Potential lässt sich beliebig formen. Natürlich ergeben nicht alle möglichen Kombinationen eine brauchbare Simulation, deswegen sind schon einige Voreinstellungen im *Pull-Down*

²Ausnahme: Potential wird konstant gewählt.

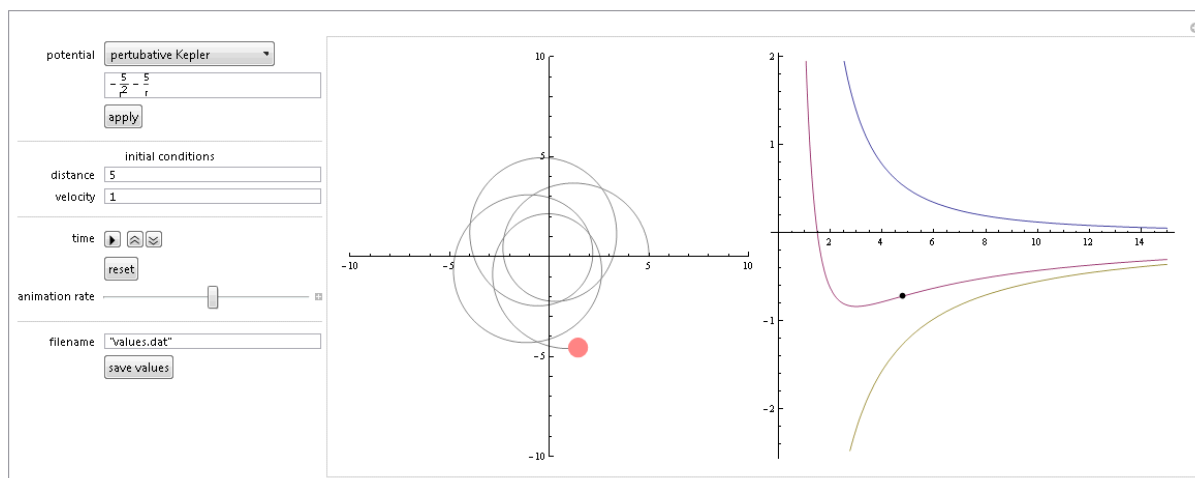


Abbildung 1: Screenshot der Anwendung mit dem Potential $V(r) = -\frac{5}{r} - \frac{5}{r^2}$.

Menü wählbar, die als Anhaltspunkte dienen können.

Achtung: hohe Potentiale (entsprechen hohen Energien) können vom Integrator nicht gehandhabt werden, da dann die Geschwindigkeiten zu groß werden. Das Gleiche gilt für effektive Potentiale, deren Minima sehr nahe im Ursprung liegen.

2 Bedienung des Applets

Die .cdf-Datei lässt sich entweder mit *Mathematica* (Version 6.0 und aufwärts) oder mit dem kostenlosen *Wolfram CDF Player* öffnen. Es kann vorkommen, dass *Mathematica* die .cdf-Datei nicht korrekt abspielt. In diesem Fall kann man einfach die .nb-Datei mit *Mathematica* öffnen. Nach dem Starten des Applets erscheint ein Fenster, welches im folgenden erläutert werden.

2.1 Benutzeroberfläche

Das Fenster ist in drei Bereiche eingeteilt. Im linken Bereich befinden sich die unten erläuterten **Bedienelemente**. Im mittleren Bereich wird die Bewegung Teilchens in einem Koordinatensystem gezeigt. Die **Trajektorie** wird durch eine graue Linie dargestellt und wird als Spur des Teilchens gezeichnet. Im rechten Bereich ist das gewählte Zentralpotential, das Zentrifugalpotential, sowie das aus beiden resultierende **effektive Potential** dargestellt. Zusätzlich wird mit einem schwarzen Punkt auf der Potentialkurve der zeitabhängige Abstand des Teilchens illustriert.

Erläuterung der Bedienelemente:

- **„potential“** Hier kann ein beliebiges zentralsymmetrisches Potential eingegeben werden. Jedoch führen nicht alle Potentiale zu stabilen Bahnen. Eine Auswahl stabiler Bahnen ist im oberen Drop-Down Menü verfügbar. **Bitte zum Uebernehmen der Aenderung nur den „apply“-Button nutzen! Mit Druecken der Entertaste wird eine neue Instanz des Programms geoeffnet.**
- **„initial conditions“** Hier lassen sich beliebige Anfangsbedingungen einsetzen. Die Werte sind in m bzw. m/s . Weiterhin lassen sich die Anfangsbedingungen durch Klicken Koordinatensystem im auswählen. Hält man die linke Maustaste gedrueckt, werden die neuen Werte angezeigt. Beim Loslassen werden diese Werte uebernommen.
- **„time“** Hier lässt sich die Simulation starten, anhalten, schneller, langsamer machen oder die Zeit zuruecksetzen.
- **„animation rate“** Hiermit lässt sich die Animationsrate, also die Anzahl der Bilder pro Sekunde einstellen.
- **„show values“/„save values“** An dieser Stelle unterscheiden sich die .cdf- und die .nb-Datei. Bei der .nb-Datei lassen sich die Werte direkt in eine Datei mit angegebenem Namen direkt in den Ordner in der sich die .nb-Datei befindet speichern. Dies ist .cdf-Datei leider nicht moeglich. Stattdessen wird die Wertetabelle anstelle der Animationen angezeigt und lässt sich aus dem Fenster heraus kopieren.

2.2 Tipps für Vorführung

Bei der Vorführung des Applets sind die **Presets** sehr hilfreich. Diese befinden sich im *Pull Down Menu* des Potentialfensters. Es stehen verschiedene vorgefertigte Potentiale mit geeignet gewählten Vorfaktoren

bereit, die man einfach auswählen kann. Nach der Auswahl wird das entsprechende Potential geladen und dargestellt.

Vorgefertigte Potentiale:

- Kepler potential $V(r) = -\frac{5}{r}$

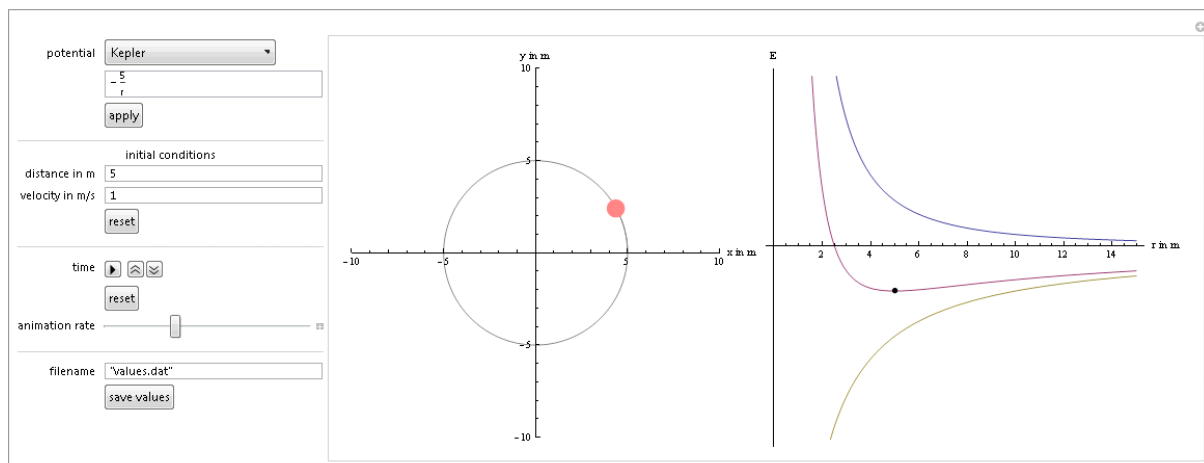


Abbildung 2: Kepler - Dient der Veranschaulichung der Bewegung der Erde um die Sonne

- Relativistic Kepler $V(r) = -\frac{5}{r} - \frac{5}{r^3}$

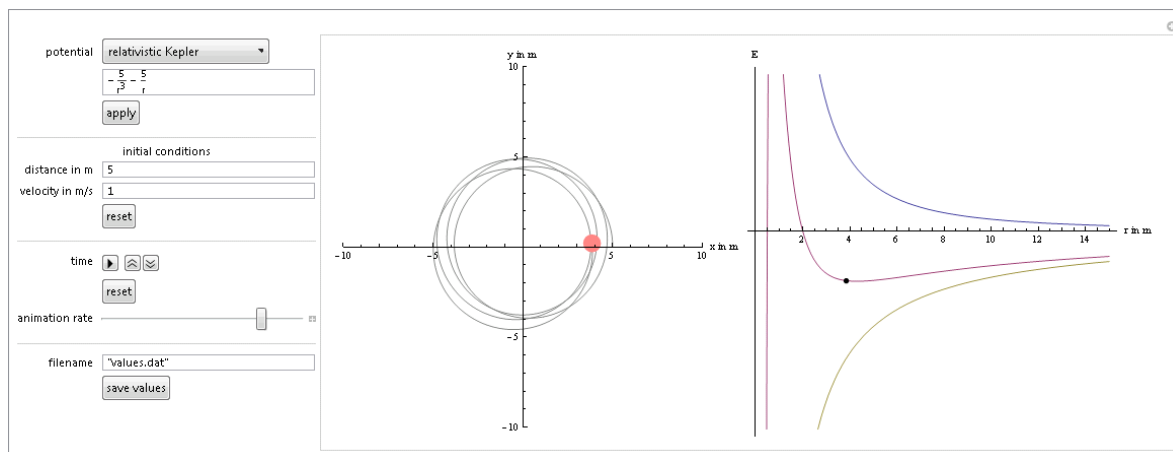


Abbildung 3: Kepler relativistisch - Der r^{-3} Term ist eine Korrektur aus der ART. Die Periheldrehung ist stark übertrieben dargestellt.

- **Modified Kepler** $V(r) = -\frac{5}{r} - \frac{5}{r^2}$

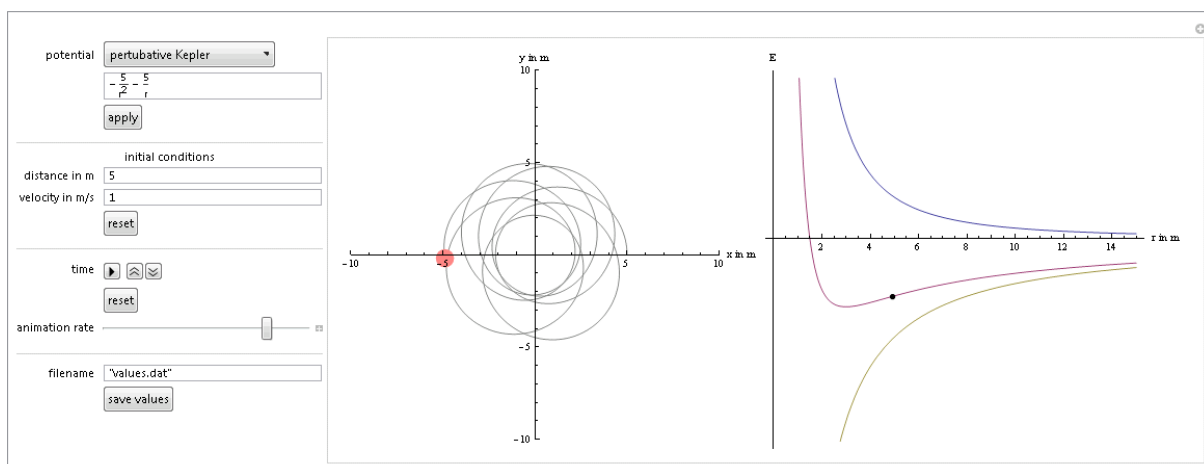


Abbildung 4: *Kepler modifiziert* - Durch Veränderung des Potentials ergibt sich eine Periheldrehung

- **Constant force** $V(r) = r$

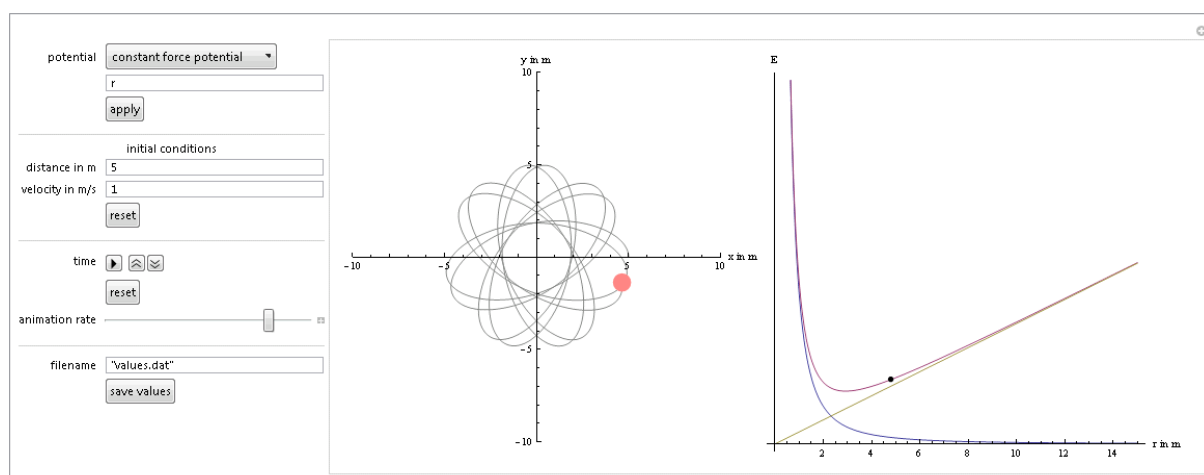


Abbildung 5: *Constant force* - Veranschaulichung des Drehimpulsanteils

- **Harmonic oscillator** $V(r) = 5 \cdot 10^{-3} \cdot r^2$

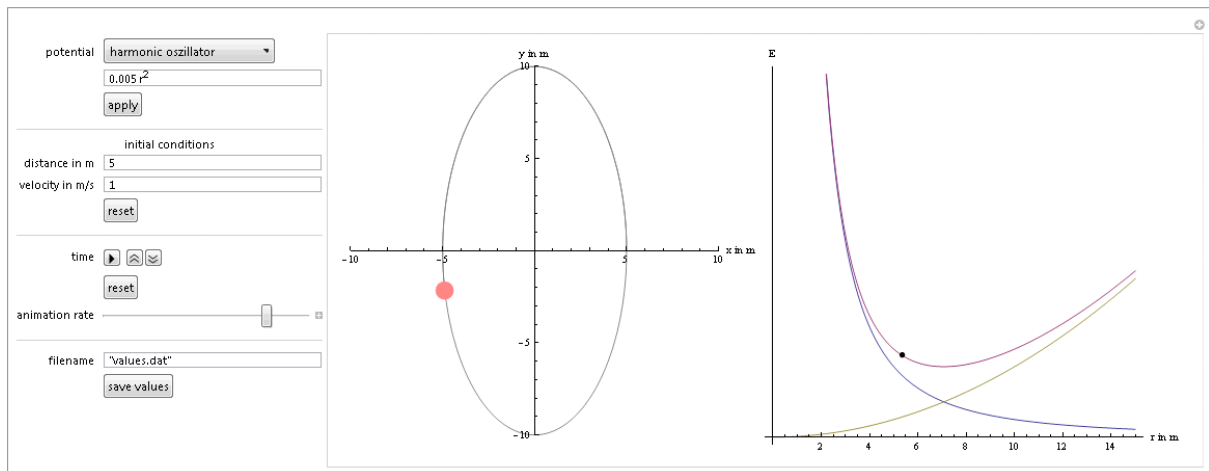


Abbildung 6: *Harmonischer oscillator* - Zwei gekoppelte Pendel jeweils in x- und in y-Richtung

- **Modified harmonic oscillator** $V(r) = -\frac{10}{r} + 0.1r^2$

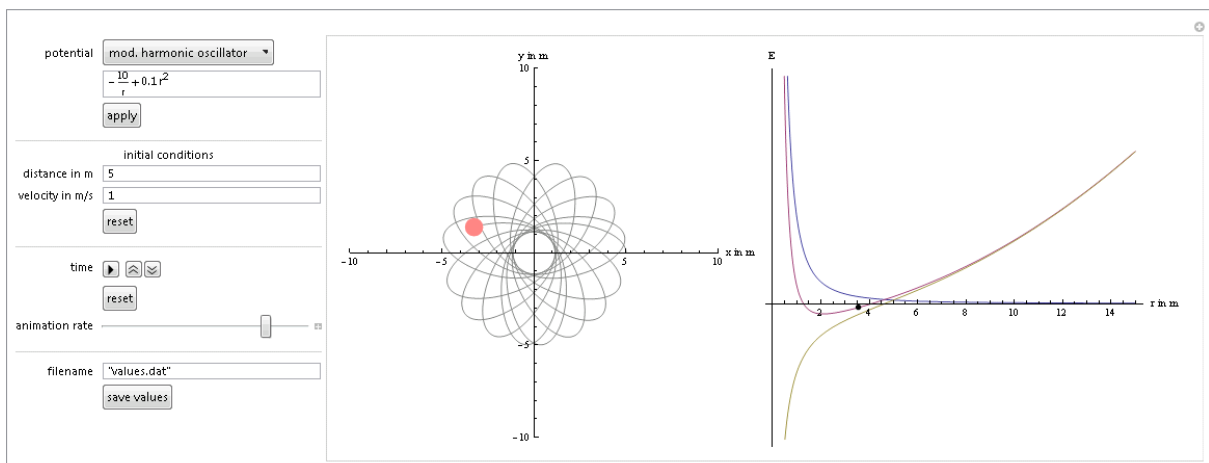


Abbildung 7: *modifizierter Harmonischer Oszillator* - Zwei gekoppelte Pendel überlagert mit dem Kepler Potential

- Repulsive potential $V(r) = \frac{5}{r}$

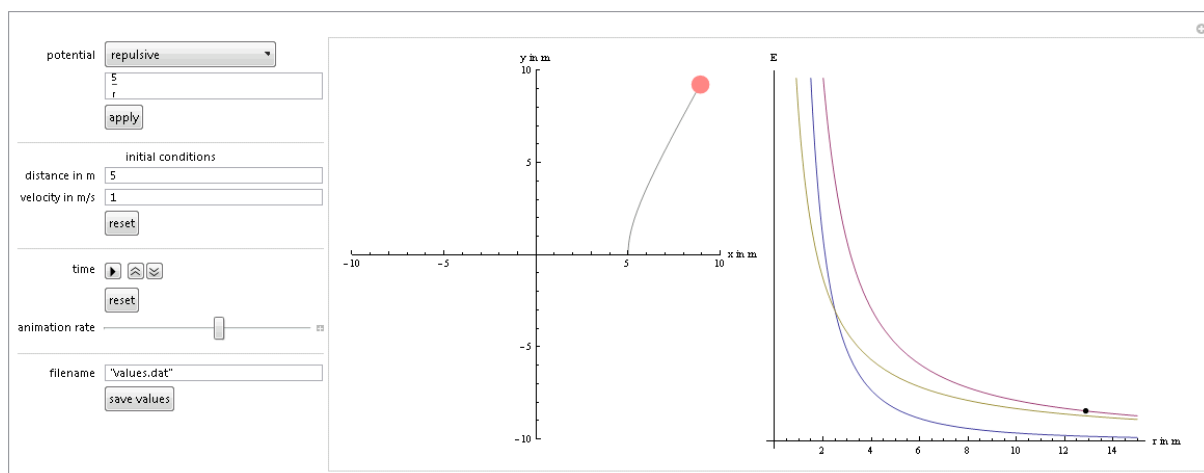


Abbildung 8: Abstoßendes Potential - Das Teilchen entfernt sich vom Kraftzentrum

- Preset 1 $V(r) = \frac{7}{r} + \frac{r^2}{70}$

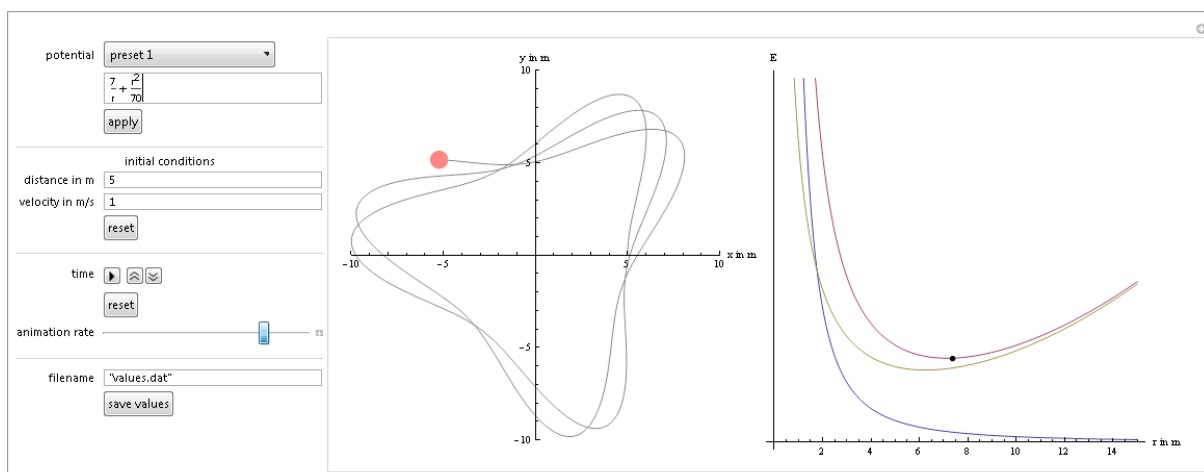


Abbildung 9: Preset 1

3 Programm

3.1 Programmstruktur

Es existieren zwei Versionen dieses Programms. Die eine Version ist eine normale *Mathematica*-Datei mit einer .nb-Endung. Diese Datei enthaelt den ausfuehrbaren Quellcode und laesst sich mit *Wolfram Mathematica* Version 6.0 oder hoeher oeffnen, abspielen und auch manipulieren. Die andere Version ist eine .cdf-Datei und laesst sich mit dem kostenlosen *Wolfram CDF Player* (Download unter <http://www.wolfram.com/cdf-player/>) abspielen. Diese Datei enthaelt nur das fertige Programm und ist nicht manipulierbar.

Quelltext des Programms

Prinzipiell besteht das Programm aus drei Teilen. Im ersten Teil wird die Differentialgleichung numerisch geloest und im zweiten Teil die Loesungen zeitlich abhaengig dargestellt. Im dritten Teil des Programms befinden sich die Steuerung der Variablen durch den Benutzer und die dazugehoerigen Bedienelemente. Der vollstaendige Quellcode befindet sich im Anhang.

Zur Loesung der Differentialgleichung wird die Mathematicafunktion *NDSolve* fuer den Bereich $0 < t < 1000$ genutzt. Diese Loesung wird in *solution* abgespeichert und ist im weiteren Programm damit abrufbar.

```
solution =
NDSolve[{
  x''[t] == -(x[t]/Sqrt[x[t]^2+y[t]^2])D[potential,r]/.r->Sqrt[x[t]^2+y[t]^2],
  y''[t] == -(y[t]/Sqrt[x[t]^2+y[t]^2])D[potential,r]/.r->Sqrt[x[t]^2+y[t]^2],
  x[0] == x0,
  y[0] == 0,
  x'[0] == 0,
  y'[0] == v0
}, {x[t], y[t], x'[t], y'[t]}, {t, 0, 1000}, MaxSteps->Infinity]
|
```

Das Potential *potential* und die Anfangsbedingungen *x0* und *v0* werden vom Benutzer im letzten Teil des Programmcodes vorgegeben. Anschließend werden Variablen zur spaeteren Nutzung definiert und der Speicherort als Ort der .nb-Datei festgelegt.

```
angularMomentSquare = 1/2 (x0*v0)^2
absoluteDistance = Sqrt[x[t]^2 + y[t]^2]
koordinates = {x[t], y[t]} /. solution
SetDirectory[NotebookDirectory[]]
```

Die Graphen werden mit dem *Plot*- bzw. dem *ParametricPlot*-Befehl realisiert. Um die Punkte in den Selben Graphen zu setzen werden diese und der zugehoerige Plot in einer *Show*-Umgebung zusammengefasst.

```

TableForm[{
  Show[
    ParametricPlot[
      Evaluate[koordinaten], {t, 0, tplot},
      PlotStyle -> Gray, PlotRange -> 2*x0, ImageSize -> 400,
      PlotPoints -> Round[tplot] + 10,
      AxesLabel -> {If[show, "x_0 in m", "x in m"],
                    If[show, "v_0 in m/s", "y in m"]}
    ],
    Graphics[{
      {PointSize[.05], Pink, Point[koordinaten /. t -> tplot]},
      Text[If[show,
        Grid[{"set new initial conditions:", SpanFromLeft, SpanFromLeft},
             {"distance =", pt[[1]], "m"}, {"velocity =", pt[[2]], "m/s"}], ""], pt
    ]
  ]
],
Show[
  Plot[
    {angularMomentumSquare/r^2, angularMomentumSquare/r^2 + potential, potential},
    {r, 0, 3*x0}, ImageSize -> 400, AspectRatio -> 1.,
    PerformanceGoal ->
      If[$ControlActiveSetting, "Speed", "Quality"],
    AxesLabel -> {"r in m", "E"}, Ticks -> {Automatic, None}
  ],
  Graphics[
    {PointSize[.015], Black, Point[{absoluteDistance,
      angularMomentumSquare/absoluteDistance^2 + potential /.
      r -> Sqrt[x[t]^2 + y[t]^2]} /. solution /. t -> tplot]}
  ]
]
]

```

Die Plots sind in einer *EventHandler*-Umgebung eingefasst, um die Wahl der Anfangsbedingungen mit der Maus zu ermöglichen. Die Mauskoordinaten werden in *pt* gespeichert und bei gedrückter linker Maustaste am Ort der Maus dargestellt. Der darauf folgende Code erstellt eine Wertetabelle und speichert sie im eingegebenem Dateinamen ab, sobald der „save“-Button gedrückt wurde.

```

If[save,
  valueTable =
    Table[Flatten[{Round[t, 0.1], x[t] /. s, y[t] /. s}], {t, 0,
      tplot, .1}]; PrependTo[valueTable, {"time", "x(t)", "y(t)"}];
  PrependTo[valueTable, {"", "v0 =", v0}];
  PrependTo[valueTable, {"", "x0 =", x0}];
  PrependTo[valueTable, {"initial conditions:", "", ""}];
  PrependTo[valueTable, {"potential:", "V(r)=", p}];
  Export[filename, valueTable]; save = False, ""
]

```

Im dritten Teil des Programms befinden sich alle Steuerelemente, die von der Umgebenden *Manipulate*-Umgebung interpretiert und fuer den gesamten Inhalt zugaenglich gemacht werden. Hier kann der Benutzer ein vorgefertigtes Potential oder ein eigenes waehlen, die Anfangsbedingungen setzen, die Zeit steuern und die Werte in eine Datei speichern. Eine detaillierte Beschreibung der einzelnen Punkte befindet sich im Quellcode im Anhang.

```

{{potential, -5/r, "potential"}, {
  -5/r -> "Kepler",
  -5/r - 5/r^2 -> "pertubative Kepler",
  -(5/r^3) - 5/r -> "relativistic Kepler",
  r -> "constant force potential",
  0.005* r^2 -> "harmonic oszillator",
  -(10/r) + 0.1*r^2 -> "mod. harmonic oscillator",
  5/r -> "repulsive",
  7/r + r^2/70 -> "preset 1"
}, ControlType -> PopupMenu},
{{p, -5 r^-1, ""}, -5 r^-1},
{{fake, None, ""}, Button["apply", fake = None] &},
Delimiter,
Item["initial conditions", Alignment -> Center],
{{x0, 5, "distance in m"}, 5},
{{v0, 1, "velocity in m/s"}, 1},
{{fake, None, ""}, Button["reset", {x0, v0} = {5, 1}] &},
Delimiter,
{{tplot, 0.001, "time"}, 0.001, 1000, ControlType -> Animator,
  AnimationRunning -> False,
  AppearanceElements -> {"PlayPauseButton", "FasterSlowerButtons"},
  AnimationRate -> animationrate},
{{tplot, 0.01, ""}, Button["reset", tplot = .01] &},
{{animationrate, 1, "animation rate"}, 1, 10},
Delimiter,
{filename, "values.dat"},
{{save, False, ""}, Button["save values", save = True] &}

```

4 Theorie (kurz)

4.1 Beschreibung mittels Differentialgleichung

Im Folgenden beschränken wir uns auf die Lösung des klassischen Zweikörperproblems mit Zentralpotential (siehe [1][2]). Die Newtonschen Bewegungsgleichungen lauten

$$\begin{aligned} m_1 \ddot{\mathbf{r}}_1 &= -\nabla_{\mathbf{r}_1} V(|\mathbf{r}_1 - \mathbf{r}_2|) \\ m_2 \ddot{\mathbf{r}}_2 &= -\nabla_{\mathbf{r}_2} V(|\mathbf{r}_2 - \mathbf{r}_1|) \end{aligned}$$

wobei das Potential im Applet die Form

$$V(r) = V(|\mathbf{r}_1 - \mathbf{r}_2|) = -G \cdot \frac{m_1 m_2}{|\mathbf{r}_1 - \mathbf{r}_2|^\alpha}$$

hat und r der Betrag des Abstandes der zwei Körper ist.

Die DGL für $\alpha = 1$ (Kepler Problem) lautet dann

$$\begin{aligned} m_1 \ddot{\mathbf{r}}_1 &= -G \frac{m_1 m_2}{|\mathbf{r}_1 - \mathbf{r}_2|^3} (\mathbf{r}_1 - \mathbf{r}_2) \\ m_2 \ddot{\mathbf{r}}_2 &= -G \frac{m_1 m_2}{|\mathbf{r}_2 - \mathbf{r}_1|^3} (\mathbf{r}_2 - \mathbf{r}_1) \end{aligned}$$

Diese wird durch den Integrator im Runge-Kutter-Verfahren mit den gegebenen Anfangsbedingungen gelöst (siehe ??).

Analytisch kann man diese Problem ebenfalls lösen. Das DGL lässt sich in Relativkoordinaten reduzieren zu

$$\begin{aligned} (I) \quad m_2 m_1 \ddot{\mathbf{r}}_1 &= -m_2 \nabla_{\mathbf{r}_1} V(|\mathbf{r}_1 - \mathbf{r}_2|) \\ (II) \quad m_1 m_2 \ddot{\mathbf{r}}_2 &= -m_1 \nabla_{\mathbf{r}_2} V(|\mathbf{r}_2 - \mathbf{r}_1|) \\ (I - II) : m_2 m_1 (\ddot{\mathbf{r}}_1 - \ddot{\mathbf{r}}_2) &= -m_2 \nabla_{\mathbf{r}_1} V(|\mathbf{r}_1 - \mathbf{r}_2|) + m_1 \nabla_{\mathbf{r}_2} V(|\mathbf{r}_2 - \mathbf{r}_1|) \\ &\Rightarrow \mu \ddot{\mathbf{r}} = -\nabla_{\mathbf{r}} V(r) \end{aligned}$$

mit $\mathbf{r} = \mathbf{r}_2 - \mathbf{r}_1$, $r = |\mathbf{r}_2 - \mathbf{r}_1| = |\mathbf{r}_1 - \mathbf{r}_2|$ und der reduzierten Masse $\mu = \frac{m_1 m_2}{m_1 + m_2}$. Wir leiten zu-

erst den Drehimpulserhaltungssatz ab. Dieser gilt für allgemeine Zentralpotentiale, weshalb wir dies in Relativkoordinaten tun.

$$\begin{aligned} \underline{L} &= \mathbf{r} \times \mathbf{p} = \mu \mathbf{r} \times \dot{\mathbf{r}} \\ \dot{\underline{L}} &= \mu (\dot{\mathbf{r}} \times \dot{\mathbf{r}} + \mathbf{r} \times \ddot{\mathbf{r}}) \\ &= \mu (\underbrace{\dot{\mathbf{r}} \times \dot{\mathbf{r}}}_{=0, da \dot{\mathbf{r}} \parallel \dot{\mathbf{r}}} + \mathbf{r} \times \ddot{\mathbf{r}}) \\ &= \underbrace{\mathbf{r} \times -\nabla_{\mathbf{r}} V(r)}_{=0, da V(r) \text{ zentralsymmetrisch} \Leftrightarrow \nabla_{\mathbf{r}} V(r) = |\nabla_{\mathbf{r}} V(r)| \mathbf{e}_r} = 0 \\ \Rightarrow \underline{L} &= const \end{aligned}$$

Die Bewegung findet also nur in der Ebene senkrecht zum Drehimpuls \underline{L} statt. Ohne Beschränkung der Allgemeinheit legen wir den Drehimpuls $\underline{L} = L \cdot e_z$ in z-Richtung. Das hat für die numerische Lösung die Folge, dass wir die Bewegung nur in der xy-Ebene berechnen brauchen. Des Weiteren verlieren wir keine Information durch die Darstellung der Bewegung in der xy-Ebene. Erwähnt sei noch, dass es auch durchaus eine Bewegung des hier nicht betrachteten Schwerpunktes in z-Richtung geben kann.

Um die Bewegung schon an ihrem Potential zu erkennen führen wir ein effektives Potential V_{eff} ein, denn dieses zeigt alle Merkmale der Bewegung auf. Dazu nutzen wir den Energieerhaltungssatz und den Drehimpuls, die wir in Polarkoordinaten umschreiben.

$$\begin{aligned} L &= \underline{\mu r} \times \dot{\underline{r}} = -\mu r^2 \dot{\phi}^2 \underbrace{e_{\vartheta}}_{, da \vartheta = \frac{\pi}{2}} = \mu r^2 \dot{\phi}^2 e_z \\ E &= \frac{1}{2} \mu \dot{\underline{r}}^2 + V(r) = \frac{\mu}{2} (\dot{r}^2 + r^2 \dot{\phi}^2) + V(r) \\ \Rightarrow E &= \frac{\mu}{2} \dot{r}^2 + \frac{L^2}{2\mu r^2} + V(r) = \frac{\mu}{2} \dot{r}^2 + V_{eff}(r) \end{aligned}$$

Das effektive Potential besitzt, wenn ein gebundener Zustand vorliegt, ein Potentialminimum. Die Schnittpunkte mit der Energie ergeben die Umkehrpunkte bzw. den minimalen und maximalen Radius der Bewegung. Zwischen diesen Punkten findet die Bewegung statt.

Die gezeigten Berechnungen lassen sich leicht für ein zusammengesetztes Potential mit N Termen wie es im Applet simuliert wird, nachvollziehen. Das simulierte Potential lautet

$$V(r) = - \sum_{i=1}^N G_i M_i \cdot r^{\alpha_i}$$

mit dem Vorfaktor $G_i M_i$ und der Ganzzahligen Potenz α_i . Im Applet wird noch eine Skalierung zur Anpassung an die Bildschirmauflösung benutzt. Dazu wird jeder Vorfaktor $G_i M_i$ mit dem Wert 10^6 multipliziert, dieser Wert ist so gewählt, dass das Kepler Problem gut dargestellt wird. Des Weiteren ist in diesen Vorfaktoren die Masse schon enthalten.

Literatur

- [1] Herbert Goldstein. *Klassische Mechanik*. Akad. Verlagsgesellschaft, Wiesbaden, (aus d. engl. übers. von günter gliemann) 7. edition, 1983.
- [2] Wolfgang Nolting. *Grundkurs Theoretische Physik 1. Klassische Mechanik*. Springer, Berlin, 2004.

Anhang

Des gesamte Mathematica-Quellcode befindet sich auf den nächsten zwei Seiten.


```

Manipulate[
  With[
    {solution =
      NDSolve[ $\left\{x''[t] = -\left(\frac{x[t]}{\sqrt{x[t]^2 + y[t]^2}}\right) D[\text{potential}, r] /. r \rightarrow \sqrt{x[t]^2 + y[t]^2}, y''[t] = -\left(\frac{y[t]}{\sqrt{x[t]^2 + y[t]^2}}\right) D[\text{potential}, r] /. r \rightarrow \sqrt{x[t]^2 + y[t]^2}, x[0] = x0, y[0] = 0, x'[0] = 0, y'[0] = v0\right\}, \{x[t], y[t], x'[t], y'[t]\}, \{t, 0, 1000\}, \text{MaxSteps} \rightarrow \infty$ ]
      (*numerical solution of the differential equation with the given potential and initial values*)

      , angularMomentumSquare =  $\frac{1}{2} (x0 * v0)^2$ , absoluteDistance =  $\sqrt{x[t]^2 + y[t]^2}$  },
    koordinates = {x[t], y[t]} /. solution; (*definition of constants and variables used below*)

    ; SetDirectory[NotebookDirectory[]]; (*set directory for saving files to the current notebook directory*)

    EventHandler[(*event handler for setting initial values using the mouse*)

    TableForm[ $\left\{\left\{\begin{array}{l} \text{Show[} \\ \text{ParametricPlot[Evaluate[koordinates], \{t, 0, tplot\}, PlotStyle \to Gray, PlotRange \to 2 * x0, ImageSize \to 400,} \\ \text{PlotPoints \to Round[tplot] + 10, AxesLabel \to \{If[show, "x_0 in m", "x in m"], If[show, "v_0 in m/s", "y in m"]\}}, \\ (*parametric plot for showing the trajectory of the particle*) \\ \text{Graphics[\{\{PointSize[.05], Pink, Point[koordinates /. t \to tplot]\}}, (*draw a point at the current position of the particle*)} \\ \text{Text[If[show, Grid[\{\{"set new initial conditions:", SpanFromLeft, SpanFromLeft\},} \\ \{\text{"distance ="}, pt[[1], "m"}, \{\text{"velocity ="}, pt[[2], "m/s"}\}], "", pt]]} \end{array}\right\}$ 

```

```

(*the text is shown when the mouse is dragged to view the picked initial conditions when the mouse is released*)
],

Show[
Plot[{angularMomentumSquare / r^2, angularMomentumSquare / r^2 + potential, potential},
  {r, 0, 3 * x0}, ImageSize -> 400, AspectRatio -> 1., PerformanceGoal -> If[$ControlActiveSetting, "Speed", "Quality"],
  AxesLabel -> {"r in m", "E"}, Ticks -> {Automatic, None}], (*show the potentials*)
Graphics[{{PointSize[.015], Black, Point[{absoluteDistance, angularMomentumSquare / absoluteDistance^2 + potential /.
  r -> Sqrt[x[t]^2 + y[t]^2} /. solution /. t -> tplot]}}] (*show the current position of the particle in the potential*)
]

, If[save, valueTable = Table[Flatten[{Round[t, 0.1], x[t] /. s, y[t] /. s}], {t, 0, tplot, .1}];
PrependTo[valueTable, {"time", "x(t)", "y(t)"}]; PrependTo[valueTable, {"", "v0 =", v0}]; PrependTo[valueTable, {"", "x0 =", x0}];
PrependTo[valueTable, {"initial conditions:", "", ""}]; PrependTo[valueTable, {"potential:", "V(r)=", p}];
Export[filename, valueTable]; save = False, ""], TableDirections -> Row, TableAlignments -> Top]

(*a table of values is constructed and saved when the save button is pushed;
a head is prepended to the values to show the settings*)

, {"MouseDown" -> ({x0, v0} = MousePosition["Graphics"], show = False, tplot = .01)},
"MouseDragged" -> ({pt = MousePosition["Graphics"], show = True})] (*end of event handler;
show new initial values when mouse is dragged and set them when mouse is released*)

],

(*the following are the control elements*)

```

```

{{potential, -5 / r, "potential"},
{-5 / r → "Kepler", -5 / r - 5 / r^2 → "pertubative Kepler",  $-\frac{5}{r^3} - \frac{5}{r}$  → "relativistic Kepler", r → "constant force potential",
0.005 * r^2 → "harmonic oszillator",  $-\frac{10}{r} + 0.1 * r^2$  → "mod. harmonic oscillator", 5 / r → "repulsive",  $\frac{7}{r} + \frac{r^2}{70}$  → "preset 1"}},
ControlType → PopupMenu}, (*a pop up menu with presets for the potential*)
{{potential, -5 r^-1, ""}, -5 r^-1}, (*to create the text field*)
{{fake, None, ""}, Button["apply", fake = None] &},
(*a fake button to prevent the user pressing the enter button (that would open a new instance of the program)*)
Delimiter, (*stripe*)
Item["initial conditions", Alignment → Center], (*text*)
{{x0, 5, "distance in m"}, 5}, (*initial distance*)
{{v0, 1, "velocity in m/s"}, 1}, (*initial velocity*)
{{fake, None, ""}, Button["reset", {x0, v0} = {5, 1}] &}, (*reset the initial values to x0=5 and v0=1*)
Delimiter, (*stripe*)
{{tplot, 0.001, "time"}, 0.001, 1000, ControlType → Animator, AnimationRunning → False,
AppearanceElements → {"PlayPauseButton", "FasterSlowerButtons"}, AnimationRate → animationrate},
(*time with play/pause, faster and slower buttons; the animation rate is set below*)
{{tplot, 0.01, ""}, Button["reset", tplot = .01] &}, (*resets the time*)
{{animationrate, 1, "animation rate"}, 1, 10}, (*sets the animation rate*)
Delimiter, (*stripe*)
{filename, "values.dat"}, (*sets the filename*)
{{save, False, ""}, Button["save values", save = True] &} (*save button*)
]

```