Graphische Programmierung der klassischen Mechanik

– Das Computerexperiment als physikdidaktisches Mittel –

Autor 1*, Autor 2[+], Autor 3*

Autor 1* (Vor- u. Nachname), Autor 2[+] (Vor- u. Nachname), Autor 3* (Vor- u. Nachname)

*Adresse Autor 1, Autor 3, +Adresse Autor 2
Korrespondenz e-mail
(Eingegangen: TT.MM.2019; Angenommen: TT.MM.2019)

Kurzfassung

Das Universitätsstudium der Physik ist schwer, weil der Studierende zusätzlich zum Inhalt gleichzeitig auch die mathematische Symbolsprache lernen muss. Diese Situation kann dadurch verbessert werden, dass in die Erklärungen der Rechenschritte symbolische Computerprogramme integriert werden. Das interaktive Arbeiten am Computer legt dabei viele Defizite im Verständnis offen und erlaubt deren Korrektur. In diesem Artikel wird eine graphische Programmiersprache vorgestellt. Sie eignet sich gut zur Erstellung von Lehrinhalten, weil Tippfehler vermieden werden können. So könnten Computerexperimente zur klassischen Mechanik im Hörsaal durchgeführt werden. Der bewusst gewählte Begriff "Experiment" wird abschließend motiviert.

Abstract

Studying university-level Physics is hard. This is because the student needs to learn the mathematical language in addition to the content. By integrating symbolic computation into the explanation of the mathematical steps, this situation can be improved upon. Interactive work on the Computer uncovers many deficiencies in understanding and allows to correct them. In his article we present a graphical programming language. It is suited for preparing lecture content, because typos can be avoided. In this way, computer experiments of classical mechanics could be conducted in the lecture room. The deliberately chosen term "experiment" is finally motivated.

1. Einleitung

Bei der Vermittlung physikalischer Theorien hängt die Verwendung mathematischer Symbole oft vom Kontext ab[1].

In [1] werden daher zur Erklärung der klassischen Mechanik Computerprogramme verwendet. Diese müssen eindeutig formuliert sein, um interpretierbar zu sein. Die dabei verwendete Programmiersprache ist Scheme [2], ein Dialekt der LISP-Sprache.

In den Lehrplänen der Physik ist das interaktive Arbeiten am Computer jedoch nur wenig verbreitet [3][2].

Naturgemäß ist das Schreiben von Computerprogrammen in der Informatik zentraler Bestandteil. In der Didaktik der Informatik werden in einem weiteren Schritt graphische Werkzeuge als Lehrmittel eingesetzt [4]. Programme werden nicht als Text eingegeben, wodurch einfache Syntaxfehler vermieden werden.

Die hier vorgestellte graphische Oberfläche wurde mittels [6] programmiert und basiert weiters auf [5], einer in einem modernen LISP Dialekt implementierten Version der in [1] verwendeten Programmbibliothek.
In [7] wird darauf hingewiesen, dass Lernvideos von den Studierenden nur bedingt angenommen werden[3], daher sei hier erwähnt, dass das System für Vorträge direkt im Hörsaal konzipiert ist, wiewohl ein Praxistest aussteht.
Das Laufenlassen der Programme wird hier bewusst als Ausführung eines Experiments bezeichnet. Die erkenntnistheoretische Bedeutung des Experiments wurde bereits in [8] aus dem Blickwinkel der Physikdidaktik beleuchtet. In diesem Artikel erfolgt der Versuch, das Erstellen von Computermodellen an diesen Begriffsrahmen anzugliedern.

[1] Aus dem Vorwort von [1]: "Symbols have ambiguous meanings and depend on context, and often even change within a given context."
[2] Aus dem Abstract von [3]: "few report using computation with interactive engagement methods in the classroom"
[3] Aus der Konklusion von [7] "The trend of students not accessing long videos completely was more prominent in lecture-oriented videos than the laboratory videos."

2. Ein einfaches Computerexperiment

Mittels der graphischen Oberfläche werden folgende Funktionen definiert:

a) Path-of-a-Free-Particle(time): Pfad eines freien Teilchens in zwei Dimensionen.

b) Kinetic-Energy(velocity): Kinetische Energie, vorerst einfach als das Quadrat der Geschwindigkeit, also für den Fall m=2.

c) Lagrangian(time, position, velocity): Lagrangefunktion des freien Teilchens.

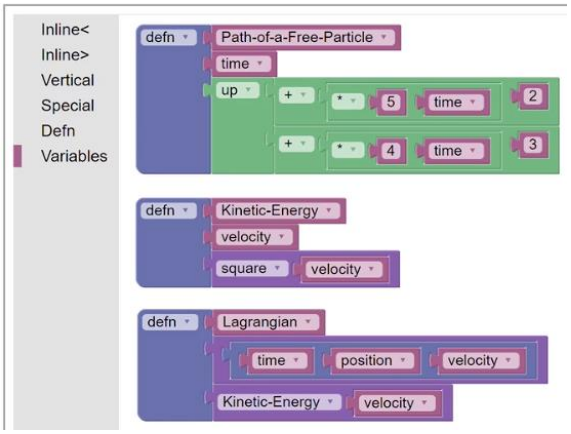

Abb.1: Die Definition der Funktionen mit graphischen Bausteinen (Kacheln).

Die Kacheln im obigen Bild werden mittels der Menüpunkte links ausgewählt. Das Menü wurde im Hinblick auf die anschauliche Modellierung der Lagrange Mechanik programmiert. So ist beispielsweise die Farbe einer Kachel durch die Anzahl der Funktionsargumente festgelegt.

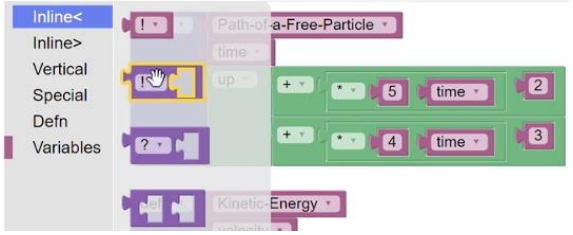

Abb.2: Auswahl einer Kachel über das Menü

Jede Kachel kann aus seiner Position herausgelöst, gelöscht oder verschoben und an einer beliebigen anderen Stelle eingefügt werden.

Die Variablen und Funktionsnamen wurden zum besseren Verständnis voll ausgeschrieben. Gewöhnungsbedürftig wird sein, dass etwa das Multiplikationszeichen * wie jede andere Funktion ganz links in der Kachel steht.

Nun wird in einem einzigen Block folgendes codiert:

a) Konstruktion der Lagrange'schen Differentialgleichungen (LDgl) aus der Lagrangefunktion.

b) Anwendung dieser LDgl auf den Pfad. Da es sich bei LDgl genau genommen um einen Operator handelt, ergibt dieser Vorgang eine neue Funktion.

c) Auswertung dieser Funktion für den Zeitpunkt t=10.

Abb.3: Konstruktion und Anwendung der Lagrange'schen Differentialgleichung in einem einzigen Block.

Da das Ergebnis der Kachel "Lagrange-equations" ein Differentialoperator ist, kann er an erster Stelle einer anderen Kachel stehen. Das Verständnis für dieses Konzept wird sicherlich etwas Zeit und Überlegung bedürfen. Durch die Rahmung der Kacheln ist die Abgrenzung der Funktionen aber ganz gut ersichtlich. Im Vergleich dazu schneidet eine rein textuelle Repräsentation schlechter ab, sie würde vielleicht wie folgt aussehen:

Lagrange-equations(Lagrangian)(Path-of-a-Free-Particle)(10)

Nun können wir das erste Computerexperiment laufen lassen. Der in die LDgl eigesetzte Pfad ist der für das freie Teilchen physikalisch richtige. Daher erhalten wir als erstes Ergebnis den Nullvektor, was auch am unteren Rand des Bildschirms angezeigt wird.

Im nächsten Schritt zeigt sich der Vorteil der graphischen Oberfläche am Deutlichsten. Die Funktion "Path-of-a-Free-Particle" wird durch einen allgemeinen Pfad ersetzt. Es handelt sich dabei um den parametrisierten Vektor ($q_x(t)$, $q_y(t)$).

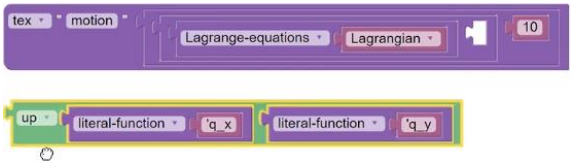

Abb.4: Der große Vorteil der graphischen Oberfläche: Ersetzen von Kacheln.

Der Ausgang dieses zweiten Computerexperiments ist sicherlich nicht jedem von vornherein klar, die Durchführung ist jedoch einfach, nämlich ein simpler Mausklick.

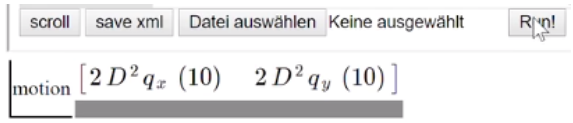

Abb.5: Das Ergebnis des zweiten Computerexperiments ist sicherlich erklärungsbedürftig.

Dieses Ergebnis ist überraschend und wird zum Staunen und Rätseln führen, es bedarf daher der genaueren Besprechung. Dem Wesen nach jedoch ist das Unverständnis vollkommen geklärt. Es besteht lediglich bezüglich der Interpretation der Formelzeichen des Ergebnisses. Denn die Rechenschritte werden

von der Software durchgeführt, um deren Verständnis es dem Experimentator gar nicht geht[4].

Die genaue Besprechung wird zum Verständnis der folgenden Punkte führen:

a) Beim Ergebnis handelt es sich um einen Vektor aus zwei Differentialgleichungen.

b) q_x ist eine Funktion mit einer einzigen Variablen als Argument.

c) D^2q_x ist die Bezeichnung für die zweite Ableitung dieser Funktion.

d) Es besteht kein Zweifel darüber, nach welcher Variablen abgeleitet wird, diese Variable besitzt hier jedoch keine Bezeichnung.

e) Im Speziellen kommt der übliche Variablenname t nirgends vor.

f) Die zweite Ableitung einer Funktion ist wieder eine Funktion einer einzigen Variablen.

Im Übrigen kann jetzt die Funktion "Path-of-a-Free-Particle" aus der Benutzeroberfläche entfernt werden.

Durch Hineinfügen einer die Masse repräsentierenden Kachel bringen wir die kinetische Energie in die korrekte Form. Dieses immer wieder neue Hinzufügen und Weglöschen in der graphischen Oberfläche zeigt deutlich, dass hier ein verbessertes Modell aus einem vorhergehenden Modell entwickelt wird.

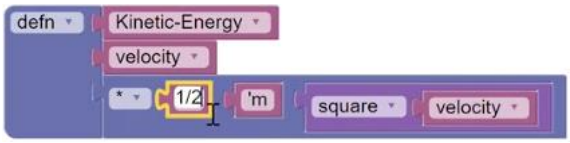

Abb.5: Verbesserung des Modelles durch Hinzufügen der Masse m zur in Abb.1 gezeigten kinetischen Energie.

Im letzten Schritt wird der Zeitpunkt 10 durch das Symbol 't ersetzt. Dadurch wird nochmals deutlich, dass der Buchstabe t ein Parameter ist und nicht etwa eine Variable, nach der differenziert wird.

Schlussendlich lassen wir das Experiment zum dritten Mal laufen und gelangen so zu den Newton'schen Gleichungen des freien Teilchens in der bekannten Form.

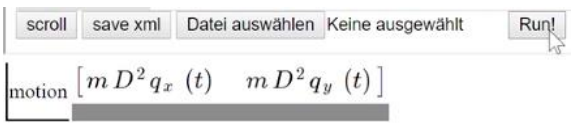

Abb.6: Das Endergebnis: die Newton'schen Gleichungen des freien Teilchens.

---

3. Zum Begriff Computerexperiment

Die Ansicht, dass der Begriff "Experiment" im Zusammenhang mit dem Abarbeiten eines Computerprogrammes adäquat ist, bedarf der Motivation. Zuallererst gibt es Parallelen zwischen einem Computerprogramm und einem Laborexperiment:

a) Der physische Rechner muss nach einem theoretischen Plan gebaut und konfiguriert werden.

b) Das Ergebnis ist vor dem ersten Ablauf unbekannt.

c) Das Computerprogramm läuft als elektronischer Schaltvorgang wie jeder andere natürliche Prozess in der Zeit ab.

d) Das Equipment vieler Laborexperimente, etwa jene zur Quantenoptik, kann als Hardware eines Rechners angesehen werden.

Die folgenden Zitate sind aus einem Artikel zur Physikdidaktik [8], in dem Computer keine Rolle spielen. Es ist bemerkenswert, wie gut die Zitate in eine Betrachtung passen, die das Computerprogramm als Experiment ansieht.

Die Abduktion ist ein ad hoc Schlussverfahren und wird wie folgt definiert:

"Ausgangspunkt einer Abduktion ist meist eine überraschende Tatsache, eine Frage oder eine Noch-Ungereimtheit, die eine Regel nahe legt, die die Frage beantwortet oder den Zweifel beseitigt. Das Staunen und Rätseln über empirische Phänomene sollte den kreativen Prozess des Begriffs- und Hypothesenbildens auslösen."

Je mehr man akzeptiert, dass das Ergebnis eines Computerprogrammes ein empirisches Phänomen, also eine gegebene Tatsache ist, desto naheliegender wird die Ansicht, dass der Dozent im zweiten obigen Beispiel dem Studenten bei genau so einer Abduktion auf die Sprünge hilft.

Akzeptiert die Studierende nun, dass der Ablauf eines Computerprogramms tatsächlich ein Experiment ist, so ist auch offensichtlich, dass es Experimente gibt, die ohne Theorie nicht denkbar sind. Und

"Demnach wäre es aus didaktischer Sicht sinnvoller, das Experiment den theoretischen Erwägungen anzuschließen."

Auch war die Anzeige des Nullvektors im ersten Programmbeispiel sehr instruktiv. Anders gesagt:

"Ausgehend von einem allgemeinen Prinzip ist durch einige mathematische Schritte eine neue physikalische Erkenntnis zutage gefördert worden."

Es erscheint recht intuitiv, dieses "Zutagefördern" mit einer speziellen Art von Experiment in Verbindung zu setzen.[5]

[4] So geht es etwa auch einem Hochenergiephysiker bei Computersimulationen zur Effizienz eines bestimmten Zählrohrtypus nicht um das genaue Verständnis der nötigen Rechenschritte [9].

[5] In [8] wird mit dem Zitat die Herleitung des snelliusschen Brechnungsgesetzes aus dem fermatschen Prinzip beschrieben. Gerade die Beschäftigung mit Computerprogrammen macht dessen Betreten

Durch die Einführung des Begriffs "Computerexperiment" bekäme die Physikdidaktik ein weiteres Mittel an die Hand, die erkenntnistheoretische Bedeutung des Experiments zu beleuchten. Denn wenn eine Theorie nicht einmal in einem derartigen Computerexperiment nachvollzogen werden kann, ist sie auf jeden Fall unphysikalisch.

Außerdem wird in einem solchen Computerexperiment anhand der Laufzeit sehr stark deutlich, dass ein Modell, welches analytische Lösungen zur Erklärung eines Phänomens bietet, vielleicht empirisch widerlegt aber nichtsdestotrotz alles andere als wissenschaftlich obsolet sein kann.

Die Studierenden werden vielmehr danach trachten, für einen gegebenen Zweck das vergleichsweise bestverfügbare Modell zu finden. Und sie lernen, dass dieses Ziel nur durch Heranziehen der Meinung anderer erreicht werden kann.

4. Zusammenfassung und Ausblick

In diesem Artikel wurde mithilfe einer graphischen Programmiersprache die Modellierung des einfachsten Falles der klassischen Mechanik vorgeführt, der Fall des freien Teilchens. Der Vorteil der dafür notwendigen präzisen Notation zeigte sich beispielsweise in der klaren Unterscheidung zwischen Variable und Parameter. Das Einsetzen von Funktionsansätzen in Differentialgleichungen war sehr schnell möglich, das Computerexperiment wurde dreimal adaptiert und laufengelassen. Dabei wurde durch die graphischen Möglichkeiten des Ersetzens von Programmteilen die graduelle Verbesserung des Modells anschaulich dargestellt. Durch die bewusste Bezeichnung dieser Vorgänge mit dem Begriff Computerexperiment wird didaktisch besonders klar, dass es kein Experiment ohne Theorie gibt und dass eine Theorie nur dann physikalisch ist, wenn sie in einem Experiment nachvollzogen werden kann.

Die dem System zugrundeliegende Programmbibliothek erlaubt die Modellierung von weiten Teilen der klassischen Mechanik. Tatsächlich wurden bereits viele weitere Beispiele innerhalb eines Notebooks als Programmtext codiert [10]. Speziell Koordinatentransformationen können im System sehr einfach durchgeführt werden, wodurch sich das getriebene Pendel für weitere Demonstrationen der graphischen Oberfläche besonders eignet.

5. Literatur

[1] Sussman, Gerald Jay; Wisdom, Jack (2014): Structure and Interpretation of Classical Mechanics, 2nd ed., Cambridge, MA: MIT Press. URL: https://mitpress.mit.edu/sites/default/files/titles/content/sicm_edition_2/book.html (Stand: 30.7.2019)

[2] Abelson, Harold; Sussman, Gerald Jay; Sussman, Julie (1996): Structure and Interpretation of Computer Programs, 2nd ed., Cambridge, MA: MIT Press. URL: https://mitpress.mit.edu/sites/default/files/sicp/index.html (Stand: 30.7.2019)

[3] Caballero, Marcos D.; Merner, Laura (2018): Prevalence and nature of computational instruction in undergraduate physics programs across the United States. In: Phys. Rev. Phys. Educ. Res. 14. URL: https://journals.aps.org/prper/abstract/10.1103/PhysRevPhysEducRes.14.020129 (Stand: 30.7.2019)

[4] Modrow, Eckart (2018): Informatik mit Snap. URL: http://ddi-mod.uni-goettingen.de/InformatikMitSnap.pdf (Stand: 30.7.2019)

[5] Smith, Colin (2019): Sicmutils in Clojure. URL: https://github.com/littleredcomputer/sicmutils (Stand: 30.7.2019)

[6] Fraser, Neil (2019): A JavaScript library for building visual programming editors. URL: https://developers.google.com/blockly (Stand: 30.7.2019)

[7] Lin, Shih-Yin et.al.: Exploring physics students' engagement with online instructional videos in an introductory mechanics course. In: Phys. Rev. Phys. Educ. Res. 13. URL: https://journals.aps.org/prper/abstract/10.1103/PhysRevPhysEducRes.13.020138 (Stand: 30.7.2019)

[8] Krause, Eduard (2017): Einsteins EJASE-Modell als Ausgangspunkt physikdidaktischer Forschungsfragen. In: Physik und Didaktik in Schule und Hochschule, Band 1, Nr. 16. S. 57-66.

[9] XXXX (1998), CERN CMS-Note

[10] XXXX (2019), URL: https://nextjournal.com

unnötig, aber die Assoziation zum Begriffsfeld "Gedankenexperiment" drängt sich auf.