

物理学の学習のためのプログラマブルな シミュレータと環境の提案

東京工業大学 情報理工学院 数理・計算科学系
18B04657 木内康介
指導教員 増原英彦教授

1 はじめに

高等学校における物理学の授業において、生徒による実験は重要である。Holubova [1] は、実験室での作業は理論的な概念を検証する最も重要な方法であり、生徒は実験を通して自分の考えを判断することができると述べている。しかし実際は、実験の実施は完全ではない。林らの調査 [2] によると、力学分野における最も基本的な「運動の法則」に関する実験の経験は、2014 年の調査時点で 60% 程度にしか満たない。理由としては、実験用の装置の準備や測定が難しいことや、実験を行うのに時間を要することが考えられる。

そこで実験の代替として考えられるのが、シミュレーションの利用である。シミュレーションを用いることで、誰でも実験と同様な学習効果を得ることができる。実際 Ajredini [3] は、実際の実験とシミュレーションで得られる知識に大きな差は無いと結論づけている。

しかし、既存のシミュレータでは実際に生徒が解く問題との間のギャップが大きい。一般的なシミュレータは、図 1 のようにそれぞれの値や計算結果を数値として表現している。このようなシミュレータでは、生徒はバックグラウンドでどのような計算がされているのかを確認できず、結果の数値が確認できるだけである。一方、生徒が解くことになる物理の問題は図 2 のような文字式ベースの計算が主流である。そのため、自分が計算した結果の文字式とシミュレーションの結果が一致しているかを確認するためにシミュレーションで利用した値を代入して計算する必要がある。

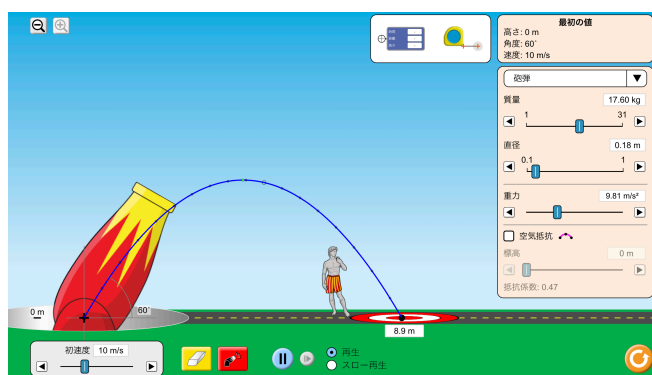


図 1 数値ベースのシミュレータ例 (PhET [4])

$$\begin{cases} ma = F & (1) \\ v = v_0 + at & (2) \end{cases}$$

$$(1) \text{ より, } a = \frac{F}{m}$$

$$(2) \text{ に代入して, } v = v_0 + \frac{F}{m}t$$

図 2 文字式ベースの計算例

そこで本研究では、物体の動きを生徒自身が文字式で定義できるシミュレータである「SimSym(Simulation with Symbols)」を提案する。SimSym では、生徒が定義した文字式に従って数値計算がなされ、シミュレーションが実行される。これにより、生徒が導出した文字式がどのような動きと対応するか簡単に確認することができる。

2 SimSym

動きの定義

SimSym では図 3 のように物体の動きを文字式で定義できる。シミュレーションを実行するためには数値が必要なので、定数の値を設定する必要がある。また、時刻を表す変数 t を自由に使用できる。

定数	ボール
$a = 4$	位置: $v_0 t + \frac{1}{2} a t^2 = 21$
$v_0 = 1$	速度: $v_0 + a t = 13$
	加速度: $a = 4$

$t = 3$

図 3 SimSym 上で物体の動きを文字式で定義する例

シミュレーションの実行

時刻 t を変化させながら物体の位置を計算し描画することで、シミュレーションが実行できる。

3 実装

実装は、フロントエンドに lively.next^{*1} を、文字式の計算に SymPy [5] を用いた。SymPy は Python のライブラリであるが、WebAssembly で実装された CPython 処理系の Pyodide^{*2} を用いることでブラウザ上で完結させた。また、数式の表示には KaTeX^{*3} を用いた。

lively.next

lively.next は Lively Kernel [6] から派生したプロジェクトで、Web プログラミング環境である。JavaScript で記述されたコンポーネントをブラウザ上で組み合わせることで、GUI アプリケーションを作成することができる。SimSym のフロントエンドは lively.next 上で作成した。

SymPy

SymPy は、文字式の計算を可能にする Python ライブラリである。文字式の定義や数値を代入しての計算などを SymPy で行っている。Python のライブラリであるが、Pyodide を用いることで lively.next から直接扱うことが可能になる。

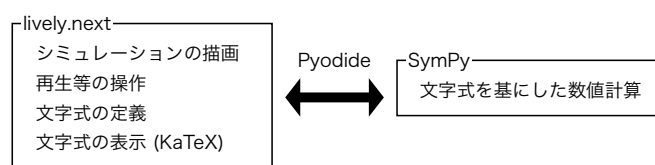


図4 SimSym の実装の概要

4 まとめと課題

本研究では文字式を基にしたシミュレーションが可能であるシミュレータ「SimSym」を提案した。SimSym を用いることで、生徒は自身が導出した文字式がどのような動きと対応するか簡単に確認することができる。結果として、物理学に対するより直感的・本質的な理解が促進できると考える。

今後の課題として、SimSym の教育効果の評価がある。評価手法として、Hake [7] が導入した normalized gain を用いる方法を提案する。満点を 100 とする pre-test と post-test の平均点をそれぞれ $\langle \text{pre-test} \rangle$, $\langle \text{post-test} \rangle$ とすると、normalized gain $\langle g \rangle$ は次のように定義される:

$$\langle g \rangle = \frac{\langle \text{post-test} \rangle - \langle \text{pre-test} \rangle}{100 - \langle \text{pre-test} \rangle}$$

高校生数十人を、

- シミュレータを利用しないグループ
- 既存のシミュレータを利用するグループ

- SimSym を利用するグループ

の 3 つに分け、各グループに共通の pre-test を課す。その後、同じ分野について同等の内容を教え、post-test を課す。各グループで normalized gain がどのような値を示すかによって、SimSym の教育効果を評価することができると考える。

参考文献

- [1] Renata Holubova. The impact of experiments in physics lessons – “why, when, how often?” . *AIP Conference Proceedings*, Vol. 2152, No. 1, p. 030007, 2019.
- [2] 林 壮一ほか. 大学生に対する高校物理実験および放射線学習の現状調査. *物理教育*, Vol. 63, No. 3, pp. 191–196, 2015.
- [3] Fadil Ajredini, et al. Real Experiments versus Phet Simulations for Better High-School Students’ Understanding of Electrostatic Charging. *European Journal Of Physics Education*, Vol. 5, No. 1, p. 59, February 2014.
- [4] Katherine K. Perkins, et al. PhET: Interactive Simulations for Teaching and Learning Physics. *The Physics Teacher*, Vol. 44, No. 1, pp. 18–23, January 2006.
- [5] Aaron Meurer, et al. SymPy: symbolic computing in Python. *PeerJ Computer Science*, Vol. 3, p. e103, January 2017. Publisher: PeerJ Inc.
- [6] Daniel Ingalls, et al. The lively kernel a self-supporting system on a web page. In Robert Hirschfeld and Kim Rose, editors, *Self-Sustaining Systems*, pp. 31–50, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.
- [7] Richard Hake. Interactive-engagement versus traditional methods: A six-thousand-student survey of mechanics test data for introductory physics courses. *American Journal of Physics - AMER J PHYS*, Vol. 66, , 01 1998.

*1 <https://lively-next.org>

*2 <https://pyodide.org/en/stable/>

*3 <https://katex.org>