

# 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、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どのような現象が起きるかを確認することができると述べている。

しかし実際は、生徒全員が実験を経験しているわけではない。力学分野において最も基本的な「運動の法則」に関する実験の実施経験は、2014 年の調査時点で 60% にしか満たない [2]。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近年利用されているのが、物理実験のシミュレーションである。シミュレーションを用いることで、実験と同様な学習効果を得ることができる [3]。

しかし、実際に生徒が解く問題の回答と、既存のシミュレータで確認できる情報の間にはギャップがある。生徒が試験などで解く物理の問題は図 1 のように物理量として文字を利用した方程式の計算を要求されることが多い。一方、例えば PhET [4] は、図 2 のように速度や質量、位置を数値のみで表現していて、どのような計算の結果その数値が導出されたかはわからない。そのため、既存のシミュレータでは生徒が求めた方程式がどのような運動を表現しているのか直接確認することができない。

そこで本研究では、物体の運動を方程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、定義した方程式に従って物体を運動させることができる。シミュレーションを実行すると、定義した方程式によって数値計算がなされ、物体の運動が可視化される。これにより、ユーザが導出した方程式がどのような運動と対応するか簡単に確認することができる。

なお、SimSym は大部分が未実装である。そのため、2 節

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

(1) より、 $a = \frac{F}{m}$  (2) に代入して、 $v = \frac{F}{m}t$

図 1 方程式の計算例

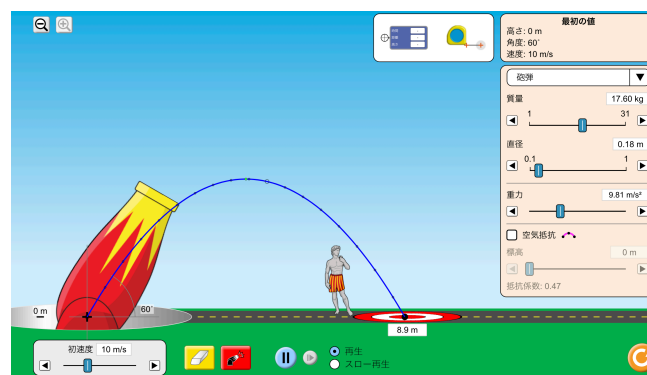


図 2 PhET のシミュレーション例

の内容は基本的にはアイデアであるが、3 に記す方針で実装することが可能であると考えている。

## 2 SimSym のアイデア

SimSym は、ブラウザ上で実行できるシミュレータである。ユーザは以下の操作を行う：

1. 物体の作成
2. 方程式の作成
3. 物体と方程式の紐付け
4. シミュレーションの実行

1. 物体の名前を入力し物体作成ボタンを押すと、その物体と描画に最低限必要なフィールド (x 座標・y 座標) が生成される。質量や速度などのフィールドは自由に追加することができる。

2. 定数・変数を自由に追加し、方程式を定義する。この際、定数と変数には値 (初期値) と次元 (長さ、質量等) を設定する必要がある。なお、時刻を表す変数  $t$  は最初から用意されている。その後、追加した定数・変数を使って方程式を定義する。長さと kilograms の和など、次元が一致していない方程式は定義できない。

3. 方程式を物体のフィールドにドラッグすると、そのフィールドに方程式を紐付けることができる。方程式とフィールドの次元が一致していない場合、紐付けることはできない。

4. シミュレーションを再生すると、時刻  $t$  が変化しながら各方程式が計算され、方程式に基づいて物体が運動する。

以下では具体例を見ていく。図 3 は、SimSym 上で斜方投射を表現した例である。この例では y 座標に紐付けている方程式が  $v_{0y}t - \frac{gt^2}{2}$  となっているため、シミュレーションを再生すると図 4 のような軌道を描く。これは、大まかに現実世界で物体を投げたときの運動と一致する。一方、例えば y 座標に紐付ける方程式を  $v_{0y}t + \frac{gt^2}{2}$  とすると、図 5 のような軌道を描き、現実世界の運動、即ち想定していた運動とは異なることがわかる。このように、方程式が表している運動をシミュレーションで確認することが可能である。

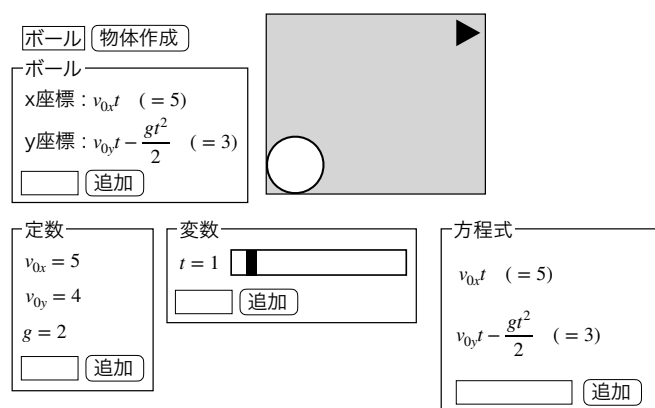


図 3 SimSym 上で斜方投射を表現した例

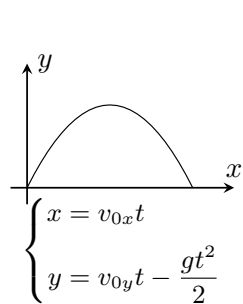


図 4 斜方投射の軌道

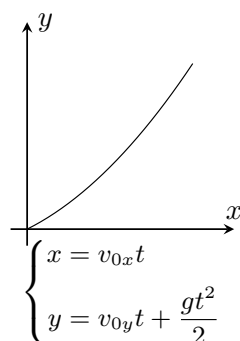


図 5 誤った軌道

### 3 実装

実装は、フロントエンドに lively.next<sup>\*1</sup> を、文字式の計算に SymPy [5] を用いる。

**lively.next**

lively.next は、JavaScript で記述されたコンポーネントを組み合わせて GUI アプリケーションを作成することができる Web プログラミング環境である。これを用いることで、ブラウザ上で簡単に GUI アプリケーションが実行できる。ユーザが物体や定数、方程式を追加するインターフェースとシミュレーションを表示する画面を lively.next で作成する。

**SymPy**

SymPy は、方程式の定義や数値代入のための Python ライブラリである。lively.next に入力された内容をもとに、

SymPy で方程式を定義し、数値計算を行っている。Python ライブラリであるが、Pyodide<sup>\*2</sup> を用いることでブラウザ上で実行させる。

#### 3.1 Pyodide

Pyodide は、WebAssembly で実装された CPython 処理系である。ブラウザ上で Python 並びにいくつかのパッケージを実行することができる。これを用いて、lively.next から SymPy を呼び出して計算することをブラウザ上だけで完結させることができる。

### 4 まとめと課題

本研究では方程式で物体の動きを定義できるシミュレータ 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確認することができる。結果として、方程式から動きを想像する力が付き、物理学に対するより直感的・本質的な理解が期待できる。

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 [6] が導入した normalized gain を用いた実験を検討する。

### 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] 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.

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

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