令和4年度 学士論文

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
学籍番号 18B04657
木内 康介

指導教員
増原 英彦 教授

令和5年2月6日

概要

ここに Abstract を書く

謝辞

本研究を進めるにあたり、増原英彦教授、叢悠悠助教に多くのアドバイスやご指導をいただきました。また、増原研究室の学生の皆様にも様々な知見や、研究におけるアドバイスをいただきました。本論文は以上の方々のご支援がなければ存在しえませんでした。この場を借りて感謝申し上げます。

目 次

第1章　はじめに

高等学校における物理学の授業において、生徒による実験は必要不可欠である。文部科学省が平成 30 年に告示した高等学校学習指導要領の理科編には以下ように記されている:

> 探究的な学習は教育課程全体を通じて充実を図るべきものであるが，観察・実験等を重視して学習を行う教科である理科がその中核となって探究的な学習の充実を図っていくことが重要である。

また、[TODO: 実験による学習効果を調査した論文を調べる]

しかし実際は、実験の実施は完全ではない。林らの調査 [1] 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。また、非常によく取り上げられる題材である「モンキーハンティング (A.1)」に関しては 10%に満たない。考えられる理由としては、モンキーハンティングを実験するためには大掛かりな装置と空間が必要になることや、測定が難しいことなどがある。

そこで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。簡単に実験ができない内容であったとしても、シミュレーションを用いれば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 [2] 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。また、シミュレーションでは実験器具の準備などの作業に割く時間を削減することができ、思考・分析・議論により多くの時間を割くことができるとも述べている。

しかし、既存のシミュレータでは不十分な点も存在する。一般的な物理学の学習用のシミュレータでは、設定されたシチュエーションにおける物体の挙動を観測することはできるが、シチュエーションそのものを大きく変化させることはできない。即ち、「この物体の座標を変更するとどう動くか」「この物体の初速度を変更するとどう動くか」「重力加速度の値を変更するとどう動くか」などといった疑問全てに対するシミュレーションは提供できない。更に、シミュレーションはその特性上数値計算をベースに実行される。一方、大学入試などにおける物理の問題は文字式の計算をベースにしている。そのため、「自分が計算した結果のこの文字式は正しく物理現象を表しているのか」ということを既存のシミュレータで確認することは難しい。

そこで、本論文ではシチュエーションを自分で設定でき、文字式をベースとしたシミュレーションが可能なシミュレータである「[TODO: ここにシミュレータの名前を入力]」を提案する。

本論文の構成は以下の通りである。第 2 章で、既存のシミュレータとそれを用いた実例

について紹介する。第 3 章で、[TODO: ここにシミュレータの名前を入力] の紹介とその効果を説明する。第 4 章で、[TODO: ここにシミュレータの名前を入力] の実現方法を説明する。第 5 章で、[TODO: ここにシミュレータの名前を入力] の評価方法を提案する。第 6 章で、まとめと今後の展望について述べる。

第2章　関連研究

この章では、既存のシミュレータとそれを用いた教育の実例について紹介する。

2.1　PhET

PhET(Physics Education Technology) [3] は、コロラド大学ボルダー校によるプロジェクトで、物理学の教育に活用できるシミュレーションの作成を目標としている。2023 年 1 月現在、ウェブサイト [4] 上では 50 以上のシミュレーションが公開されている。また、物理学のみならず化学・数学・生物学・地球科学などのシミュレーションも公開されている。

2.1.1　PhET を用いた実例

Prima [5] は、インドネシアの中学校の生徒に PhET を用いて太陽系について教える実験を行った。Prima は、PhET を利用する効果を N-Gain(normalized gain) を用いて評価している。満点を 100 とする pre-test と post-test の平均点をそれぞれ $\langle$pre-test$\rangle$, $\langle$post-test$\rangle$ とすると、 N-Gain $\langle g \rangle$ は以下のように求められる:

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

また、テストは Bloom's Taxonomy に基づき Remembering, Understanding, Applying, Analyzing の 4 領域で行われた。結果は表 2.1 の通りである。

Cognitive level	Control Group			Experiment Group		
	pretest	posttest	N-Gain	pretest	posttest	N-Gain
Remembering (C1)	33.33	76.19	0.64	44.44	84.12	0.71
Understanding (C2)	38.62	57.67	0.31	51.85	75.66	0.49
Applying (C3)	54.76	80.95	0.57	69.04	92.85	0.76
Analyzing (C4)	56.34	75.39	0.43	57.14	80.15	0.53

表 2.1: Prima による実験の結果

この調子で書いていくと [5] の内容を翻訳するだけになるので一旦中断

[6]

2.2　Scratch

Scratch [7] について
Scratch を用いた実例の紹介 [8]

第3章　提案する内容

- シミュレータの概要
- 既存のシミュレータとの差異

第4章　実装

実装は、フロントエンドに lively.next [9] を、文字式の計算に SymPy [10] を用いた。また、 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 [11] 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 [12] を用いた。

4.1　lively.next

lively.next [9] は、Lively プロジェクト [13] の一つで、ブラウザ上でプログラミングができる環境である。シミュレーション画面とそれぞれの値、SymPy を用いた計算用の画面を lively.next 上で実装している。

4.2　SymPy

SymPy [10] は、文字式の計算を可能にする Python ライブラリである。図 4.1 に簡単な例を載せる。SymPy を用いることで、実際に物理の問題を解くように文字式の計算を実行することができる。数値の代入も可能なため、シミュレーションを実行する際は SymPy で表現された式に数値を代入して計算している。

> 初速 v_0 ,加速度 $-g$ で動いている物体の時刻 t における速度と位置を求めよ

In [ ]:
```
from sympy import *
# 文字を作る
v0, g, t = symbols('v_0 g t', real=True, positive=True)
v, x = symbols('v x')
```

In [ ]:
```
# 立式する
eq1 = Eq(v, v0 - g * t)
eq1
```

Out[ ]: $v = -gt + v_0$

In [ ]:
```
eq2 = Eq(x, v0 * t - Rational(1,2) * g * t ** 2)
eq2
```

Out[ ]: $x = -\frac{gt^2}{2} + tv_0$

In [ ]:
```
# 他の文字について解くこともできる
solve(eq1, t)[0]
```

Out[ ]: $\frac{-v + v_0}{g}$

図 4.1: SymPy を用いた文字式の計算例

4.3 Pyodide

Pyodide [11] は、Mozilla が開発している WebAssembly で実装された CPython 処理系である。ブラウザ上で Python コードを実行できるほか、メジャーなパッケージにも対応している。 SymPy も Pyodide で実行可能である。実際に Pyodide で SymPy を実行する計算の例を図 4.2 に、結果を KaTeX を用いて表示した例を図 4.3 に記す。

```
<!DOCTYPE html>
<head>
  <script src="https://cdn.jsdelivr.net/pyodide/v0.22.0/full/pyodide.js"></script>
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/katex@0.16.4/dist/katex.min.css"
      integrity="sha384-vKruj+a13U8yHIkAyGgK1J3ArTLzrFGBbBc0tDp4ad/EyewESeXE/Iv67Aj8gKZ0"
      crossorigin="anonymous">
  <script defer src="https://cdn.jsdelivr.net/npm/katex@0.16.4/dist/katex.min.js" integrity="
      sha384-PwRUT/YqbnEjkZO0zZxNqcxACrXe+j766U2amXcgMg5457rve2Y7I6ZJSm2A0mS4" crossorigin="
      anonymous"></script>
  <script defer src="https://cdn.jsdelivr.net/npm/katex@0.16.4/dist/contrib/auto-render.min.js"
      integrity="sha384-+VBxd3r6XgURycqtZ117nYw44OOcIax56Z4dCRWbxyPt0Koah1uHoK0o4+/RRE05"
      crossorigin="anonymous" onload="renderMathInElement(document.body);"></script>
</head>
<body>
  <script>
    async function main() {
      let pyodide = await loadPyodide();
      await pyodide.loadPackage("sympy");
      let code = `
      from sympy import symbols, Eq, solve, latex
      v0, g, t = symbols('v_0 g t', real=True, positive=True)
      v = symbols('v')
      eq1 = Eq(v, v0 - g * t)
      latex(solve(eq1, t)[0])
      `;
      let result = pyodide.runPython(code);
      let resultDiv = document.getElementById("result");
      resultDiv.textContent = `\\[${result}\\]`;
      renderMathInElement(resultDiv);
    };
    main();
  </script>
  <p>結果</p><div id="result" style="float:␣left"></div>
</body>
```

図 4.2: Pyodide を実行する例

結果

$$\frac{-v + v_0}{g}$$

図 4.3: 実行結果

第5章　評価手法

[14] の手法を参考にして考える。

第6章　まとめと展望

参考文献

[1] 林 壮一, 川村 康文, 村上 聡. 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.

[2] Fadil Ajredini, Neset Izairi, and Oliver Zajkov. 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.

[3] Katherine K. Perkins, Wendy K. Adams, Michael Dubson, Noah D. Finkelstein, Sam Reid, Carl E. Wieman, and Ron LeMaster. Phet: Interactive simulations for teaching and learning physics. *The Physics Teacher*, Vol. 44, pp. 18–23, 2006.

[4] University of Colorado. Phet: Free online physics, chemistry, biology, earth science and math simulations. `https://phet.colorado.edu`.

[5] Eka Cahya Prima, Aldia Ridwani Putri, and Nuryani Rustaman. Learning solar system using PhET simulation to improve students' understanding and motivation. *Journal of Science Learning*, Vol. 1, No. 2, p. 60, March 2018.

[6] Nadia Rehman, Wanlan Zhang, Amir Mahmood, and Faiz Alam. Teaching physics with interactive computer simulation at secondary level. *Cadernos de Educação Tecnologia e Sociedade*, Vol. 14, No. 1, p. 127, March 2021.

[7] Massachusetts Institute of Technology. Scratch - imagine, program, share. `https://scratch.mit.edu`.

[8] Víctor López and María Isabel Hernández. Scratch as a computational modelling tool for teaching physics. *Physics Education*, Vol. 50, pp. 310 – 316, 2015.

[9] lively.next. `https://lively-next.org`.

[10] SymPy Development Team. Sympy. `https://www.sympy.org/en/index.html`.

[11] Pyodide contributors and Mozilla. Pyodide. `https://pyodide.org/en/stable/`.

[12] Katex – the fastest math typesetting library for the web. `https://katex.org`.

[13] Lively kernel. `https://www.lively-kernel.org/index.html`.

[14] Zdeněk Pucholt. Effectiveness of simulations versus traditional approach in teaching physics. *European Journal of Physics*, Vol. 42, No. 1, p. 015703, nov 2020.

付 録A　補足

A.1　モンキーハンティング

以下のような形式の問題を総称してモンキーハンティングと呼ぶ。

> 小球を、位置 $(0,0)$ から初速 v_0、 仰角 θ で発射したところ、位置 (l,h) から自由落下してくる物体に衝突した。 $\tan\theta$ の満たすべき条件を求めよ。ただし、重力加速度の大きさを g とする。