

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	1
第2章	関連研究	2
2.1	PhET . . . . .	2
2.1.1	PhET を用いた実例 . . . . .	2
2.2	Scratch . . . . .	3
第3章	提案する内容	4
第4章	実装	5
4.1	lively.lext . . . . .	5
4.2	SymPy . . . . .	5
4.3	Pyodide . . . . .	7
第5章	評価手法	10
第6章	まとめと展望	11

# 第1章 はじめに

書くこと

- 物理教育におけるシミュレーションの位置付け
- 物理教育でのシミュレーションの実例
- 現在のシミュレーションでは自分の目的に適していないことの説明

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、[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、[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、[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

### 2.1 PhET

PhET(Physics Education Technology) [1] 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 [2] 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。

ここからは、PhET の物理学シミュレーションに的を絞ってより詳しく紹介する。PhET では、学習レベルに応じたシミュレーションの分類も行っている。各レベル毎のシミュレーションの数は表 2.1 の通りである。

表 2.1: PhET における GRADE LEVEL 毎のシミュレーション数

GRADE LEVEL	個数
Elementary School	23
Middle School	37
High School	49
University	47

#### 2.1.1 PhET を用いた実例

Prima [3] 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  $\langle \text{pre-test} \rangle$ ,  $\langle \text{post-test} \rangle$  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。

この調子で書いていくと [3] の内容を翻訳するだけになるので一旦中断

[4]

## 2.2 Scratch

Scratch [5] について

Scratch を用いた実例の紹介 [6]

## 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` [7] を、文字式の計算に `SymPy` [8] 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` [9] 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` [10] を用いた。

### 4.1 `lively.next`

`lively.next` [7] は、`Lively` プロジェクト [11] 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` [8] 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 [9] 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。

---

```
1 <!DOCTYPE html>
2 <head>
3   <script src="https://cdn.jsdelivr.net/pyodide/v0.22.0/full/pyodide.js"></script>
4   <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">
5   <script defer src="https://cdn.jsdelivr.net/npm/katex@0.16.4/dist/katex.min.js" integrity="
      sha384-PwRUT/YqbnEjkZ00zZxNqcxACrXe+j766U2amXcgMg5457rve2Y7I6ZJSm2A0mS4" crossorigin="
      anonymous"></script>
6   <script defer src="https://cdn.jsdelivr.net/npm/katex@0.16.4/dist/contrib/auto-render.min.js"
      integrity="sha384-+VBxd3r6XgURycqtZ117nYw4400cIax56Z4dCRWbxyPt0Koah1uHoK0o4+/RRE05"
      crossorigin="anonymous" onload="renderMathInElement(document.body);"></script>
7 </head>
8 <body>
9   <script>
10     async function main() {
11       let pyodide = await loadPyodide();
12       await pyodide.loadPackage("sympy");
13       let code = `
14       from sympy import symbols, Eq, solve, latex
15       v0, g, t = symbols('v_0 g t', real=True, positive=True)
16       v = symbols('v')
17       eq1 = Eq(v, v0 - g * t)
18       latex(solve(eq1, t)[0])
19       `;
20       let result = pyodide.runPython(code);
21       let resultDiv = document.getElementById("result");
22       resultDiv.textContent = `\\[${result}\\]`;
23       renderMathInElement(resultDiv);
24     };
25     main();
26   </script>
27   <p>結果</p><div id="result" style="float:left"></div>
28 </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

[12] 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] 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.
- [2] University of Colorado. Phet: Free online physics, chemistry, biology, earth science and math simulations. <https://phet.colorado.edu>.
- [3] 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.
- [4] 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.
- [5] Massachusetts Institute of Technology. Scratch - imagine, program, share. <https://scratch.mit.edu>.
- [6] Víctor López and María Isabel Hernández. Scratch as a computational modelling tool for teaching physics. *Physics Education*, Vol. 50, pp. 310 – 316, 2015.
- [7] lively.next. <https://lively-next.org>.
- [8] SymPy Development Team. Sympy. <https://www.sympy.org/en/index.html>.
- [9] Pyodide contributors and Mozilla. Pyodide. <https://pyodide.org/en/stable/>.
- [10] Katex – the fastest math typesetting library for the web. <https://katex.org>.
- [11] Lively kernel. <https://www.lively-kernel.org/index.html>.
- [12] Zdeněk Pucholt. Effectiveness of simulations versus traditional approach in teaching physics. *European Journal of Physics*, Vol. 42, No. 1, p. 015703, nov 2020.