

平成 29 年度 卒業研究

物理エンジンを用いた大学編入学受験者向け
力学演習サイトの開発

釧路工業高等専門学校 電子工学科 第 44 期

浅水研究室 越野 雄大

目次

1章. 序論	3
1.1 研究背景	3
1.1.1 大学編入学制度の規模	3
1.1.2 編入学試験の物理が抱える現状	3
1.1.3 力学における「運動の想像」	4
1.2 研究目的	5
1.3 本論文の構成	5
2章. 提案手法	6
2.1 はじめに	6
2.2 本研究の提案手法	6
2.3 物理の解説サイト実例	6
3章. 研究内容	8
3.1 はじめに	8
3.2 開発環境	8
3.2.1 MathJax	8
3.2.2 p5.js	9
3.2.3 matter.js	11
3.2.4 留意点	13
3.3 サイト概要	14
3.4 試験的に作成した系	20
3.5 運動の描画	22
3.5.1 物理エンジンを用いる利点	22
3.5.2 Verlet integration	23
3.5.3 速度の定義に関する問題点	24
4章. 結論	26
4.1 はじめに	26
4.2 評価アンケート	26
4.3 総括と今後の課題	27

謝辞	28
参考文献	29
付録	30
問題 1.7.1 半円筒面上の束縛運動	30
問題 1.7.2 振り子の運動	35

第 1 章

序論

1.1 研究背景

本節では研究背景を述べる．1.1.1 項で大学編入学制度の規模に言及し，1.1.2 項では試験科目の 1 つである物理について，1.1.3 項ではその中でもさらに力学に注目して説明を行う．

1.1.1 大学編入学制度の規模

『平成 29 年度学校基本調査』[1]によると，大学の 1 年次入学者数は 629,733 人に及ぶ．対して編入学者数は約 1/100 の 7,982 人となっており，一般的な大学入試に比べて編入学制度の規模が圧倒的に小さいことが分かる．さらに入学者の出身内訳をみると，高等専門学校(以下高専)からの編入学者は 2,478 人とさらに人数が絞られる(表 1.1)．しばしば「大学入試のもう 1 つの方法」などと表現されることのある編入学制度だが，その受験者は上記に示した規模の違いに伴う深刻な「教科書不足」の問題を抱えた状態で試験勉強を進めていかなければならない．

表 1.1：平成 29 年度における入学者数の比較

	出身内訳	入学者数[人]
編入学	短期大学	4,094
	高等専門学校	2,478
	専修学校（専門過程）	1,410
	計	7,982
1 年次入学	国公立 合計	629,733

1.1.2 編入学試験の物理が抱える現状

高専生が受験する理工系の学部の試験科目は大抵の場合，数学，物理，化学，英語，専門科目の組み合わせで構成される．先に述べた「教科書不足」の現状にも科目によって大きな差が生じている．

専門科目と一般科目の違いに注目する．専門科目の分野は受験する学校や学部によって様々だが，基本的にはどれも大学に入学してから扱われる内容である．したがって大学 1・2 年の基礎的な範囲から出題される編入学試験は，高等学校（以下高校）に相当する学年でこれらを学び始めている高専生にとって有利であり，学校で使用している教科書をそのまま試験勉強に使えるケースも多い．対して一般科目では高校～大学 1・2 年に渡る全ての範囲からの出題が許されるため，その全てを網羅する形で試験勉強をすることになり，場合によっては高校生向けの教科書を読む必要も生じる．

さて，その一般科目の中でも教科書選びが最も困難であると考えられるのが本研究の取り上げる物理である．物理は本来，現象を数学で記述する学問であるため，テイラー展開，オイラーの公式，重積分，微分方程式，ベクトル解析といった数学的知識を用いて説明される．大学物理の教科書は読者がこれらの数学を理解していることを前提に書かれている（補章として数学の解説を掲載しているものもある）．対して高校においては，『高等学校学習指導要領解説 数学編』[2] が上記のいずれも含んでいないことを理由に，物理の教科書でもこれらの知識を用いることを避けた言わば「遠回し」な説明がなされている．このように大学物理と高校物理はいくつかの数学的知識の導入を境にはっきりと線引きされているため，全く別の科目であると言っても過言ではない．したがって大学物理から出題される編入学試験の受験者は，「基礎から取り掛かろうにも数学を使わない高校の教科書では全く歯が立たず，かといって大学の教科書も 1 人で読み進めるには難しすぎる」といった複雑な状況に直面することになる．この問題を解決すべく編入学試験に特化した教科書の登場が強く望まれている訳だが，数学では比較的多く出版されている「大学編入学」を謳った書籍も物理に関しては一切出回っていないのが現状である（2018 年 2 月現在）．

1.1.3 力学における「運動の想像」

ここまで述べてきた状況を改善するために，編入学受験者に向けた物理の教材を作成することが本研究の目的である．そこで，物理の中で最も基本的な分野であるニュートン力学（以下力学）を取り上げることにした．力学の学習を進める上で避けて通れないのが，式から実際の運動を想像する過程である．これは数学を用いる多くの理工学分野で同様に重要視されるため力学で習慣づけておくべき過程であるが，個人の感覚に依存してしまうために試験勉強の進度を大きく左右する要因となり得る．

1.2 研究目的

本研究が目的とするのは以下の 2 点である．

1. 大学編入学試験に特化した物理教材の作成

2. 想像力の差によって力学の進度が左右される問題の軽減

1 点目を受けて，教材を公開する媒体には Web サイトを採用する．Web サイトは PC やスマートフォン等の端末があれば誰でも閲覧可能であり，コンテンツの追加や更新も容易に行える．また次に述べる「運動を見せる」ことについても JavaScript で書かれたプログラムであれば専用ソフトのダウンロードを必要とせずブラウザ上で実行が可能である．

2 点目に関して，如何にして読者に正しい運動をイメージさせるかは従来の教科書もそれぞれに工夫を凝らしている部分である．しかし良書と言われている書籍でも紙媒体である以上はそれを読み理解するまでに多くの時間を要する．そこで本研究では実際に「運動を見せる」ことによってユーザの考察を補助できるような教材を目指す．

1.3 本論文の構成

本論文の構成を説明する．第 2 章では提案手法を述べ，本研究を始めるにあたり参考にした物理の解説サイトを従来手法として紹介する．次に第 3 章を研究内容と題し，開発環境，サイトの概要，試験的に作成したが公開には至らなかった作成内容の紹介，運動の描画について，というように説明を行う．最後に第 4 章では評価のために実施したアンケートの結果を示し，本研究の成果と今後の課題についてまとめる．また，付録として完成した 2 つの解説ページの間図部分のソースコードを掲載する．

第 2 章

提案手法

2.1 はじめに

本章では、2.2 節で提案手法を述べ、2.3 節で従来手法として物理の解説サイトの事例を紹介する。

2.2 本研究の提案手法

第 1 章で述べた本研究の目的を再掲する。

1. 大学編入学試験に特化した物理教材の作成
2. 想像力の差によって力学の進度が左右される問題の軽減

これらの目的を達成するために本研究が提案する手法は「動く図で運動を確認できる力学の e ラーニングサイト」である。サイトへの実装を目指す具体的な機能は以下の通りである。

- A) サイト内で取り上げるすべての例題・練習問題の解答に対し、運動の様子を実際に描画する機能
- B) 解に含まれる変数の値を自由に変更できるボタン・スライダー
- C) Web ページの特徴を生かした表現方法により、従来の紙媒体よりも使い易いレイアウト

2.3 物理の解説サイト事例

本研究の従来手法として、物理の解説記事を Web 上に公開しているいくつかのサイトを紹介する。

「EMAN の物理学」[2]

広江克彦氏が公開するサイト。喋り口調で書かれ、著者の考察段階も文章に含まれているのが特徴的。「趣味で物理学」シリーズとして出版もされている。フーリエ級数について解説したページの「フーリエ級数実演」というウェブアプリ(図 2.1)が非常に良くできている。

「物理のかぎしっぽ」 [3]

崎間公久が物理学とコンピュータの勉強ノートとして運営していたサイトが前身である。「物理のかぎプロジェクト」としてプロジェクト化され、多くのメンバーにより共同執筆されている。

「KIT 物理ナビゲーション」 [4]

金沢工業大学 数理工教育研究センターにより立ち上げられた物理の学習支援サイト。「KIT 数学ナビゲーション」 [5]というサイトが先に存在しており、これの物理版として新たに公開された。力学>>物理シミュレーションのメニューから閲覧可能なシミュレーションアプリ(図 2.2)が本研究の目指すものに近い。

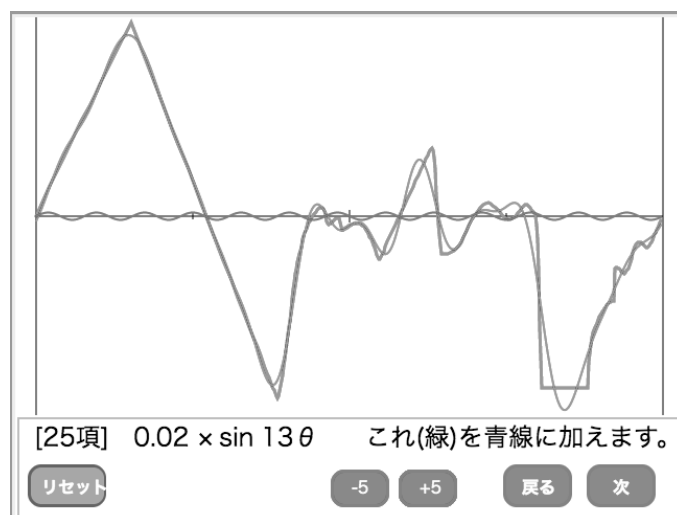


図 2.1 : EMAN の物理学より「フーリエ級数の実演」

単振動

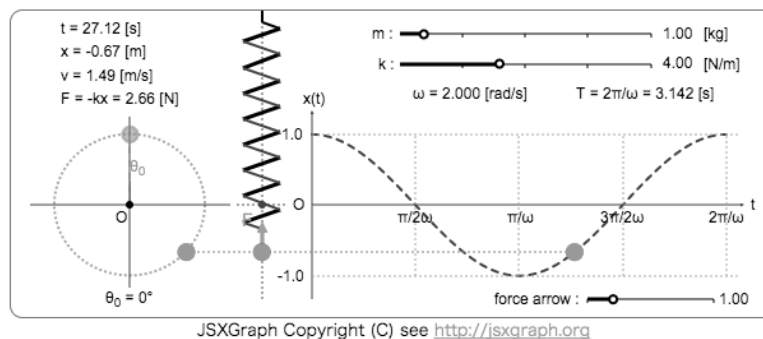


図 2.2 : KIT 物理シミュレーションより「単振動」

第 3 章

研究内容

3.1 はじめに

本章では、3.2 節で開発環境を説明する．その後 3.3 節では実際に作成したサイトの概要を、3.4 節では試験的に作成した系を紹介する．3.5 節では特に運動の描画に関して注目し考察を行う．

3.2 開発環境

本節では、本論文の読者が同様の環境で作業できるように、使用した 3 つの JavaScript のライブラリの導入手順と Hello World のプログラムを掲載する．また、3.2.4 項ではこれらのライブラリを利用する上での留意点について言及する．

3.2.1 MathJax

[概要]

MathJax[6]はページに数式を埋め込めるようにするための Javascript ライブラリである．このライブラリを宣言すれば、HTML ファイル内に直接 LaTeX の書式で数式を書き込むことができる．MathJax を用いて書かれたページを見の場合も閲覧者側は何も必要としない．

[導入から Hello World まで]

HTML の<head>タグ内に以下のソースコードを書き込むことで MathJax ライブラリを宣言できる．

```
<script type="text/javascript" async src="https://cdnjs.cloudflare.com/ajax/libs/mathjax/2.7.1/MathJax.js?config=TeX-AMS_CHTML"></script>
```

直後に以下の文を書き加えると数式を左揃えにして全角 2 文字分程度のインデントを設定できる(デフォルトでは中央揃え)．

```
<script type="text/x-mathjax-config">MathJax.Hub.Config({displayAlign: "left", displayIndent: "2em" });</script>
```

Hello World として例文を掲載する．以下の文を<body>タグ内の適当な位置に書き込み数式が表示されれば正しく読み込みができています．**MathJax** の導入に関して，詳しい解説には[7]，[8]を参照されたい．

<p>

$\frac{\text{Hello}}{\text{World}}$

オイラーの公式は $e^{i\theta} = \cos\theta + i\sin\theta$ です

</p>

3.2.2 p5.js

[概要]

p5.js[9]は Processing[10]の機能を JavaScript に移植したライブラリである．Processing とは視覚的な表現(イメージの生成，アニメーション，インタラクション)を容易に行うために開発されたプログラミング言語である．

[導入から Hello World まで]

トップページから図 3.1 のように進み，ファイル「p5.js」をダウンロードする．

p5.js

Download * Start * Reference * Libraries * Learn * Community



図 3.1 : p5.js のダウンロード方法

ダウンロードが完了したらファイルを HTML ファイルと同じ階層にコピーし，<head>タグ内に次を書き込む．

```
<script src="p5.js"></script>
```

以上でライブラリの宣言は完了である．続けて **Hello World** の手順を示す．同じ階層に `helloworld.js` という名前で新規に **JavaScript** ファイルを作成し，**HTML** ファイル `<body>` のタグ内の最後に以下を書き込む．

```
<script src="helloworld.js"></script>
```

`helloworld.js` を開き，次のように入力する．黒い四角の中央に文字が表示されれば正しく実行されている．

```
var i = 0;
function setup()
{
    createCanvas(300, 300);
}
function draw()
{
    background('black');
    stroke('white');
    fill('white');
    text('Hello World', 120, 120);
    ellipse(169, i, 6);
    if(i<118) i++;
}
```

3.2.3 matter.js

[概要]

matter.js[11]は Web ページ内で物理演算によるアニメーションを実現できる 2 次元の物理エンジンである。本研究ではこの **matter.js** を用いて力学の系を作成して問図を動かしている。

[導入から Hello World まで]

matter.js にも描画機能は存在するが本研究では物理演算のみを担当させ、描画には 3.2 節で紹介した **p5.js** を使用している。この方法は **Daniel Shiffman** 氏が YouTube チャンネル「The Coding Train」[12]で公開している **matter.js** のチュートリアル動画に基づくため詳しくはそちらを参照されたい。トップページから図 3.2 のように進み、ファイル「**matter.js**」をダウンロードする。

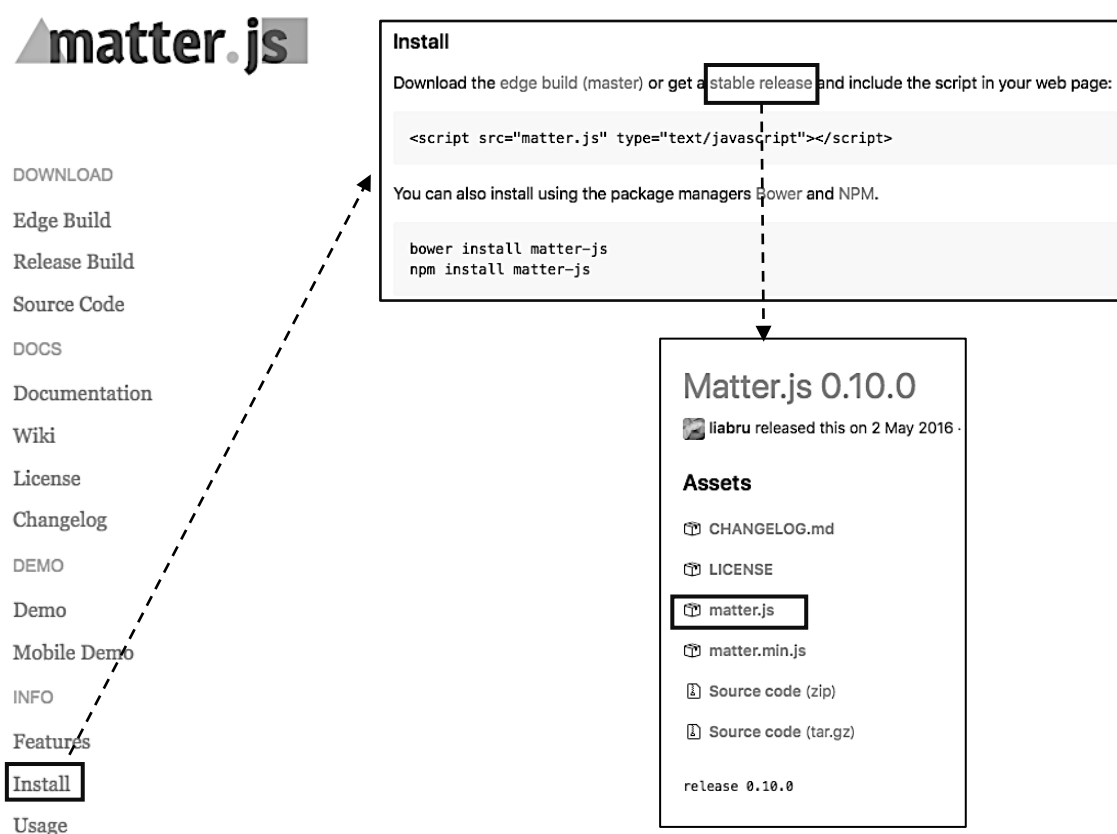


図 3.2 : **matter.js** のダウンロード手順

ダウンロードが完了したらファイルを HTML ファイルと同じ階層にコピーし、`<head>` タグ内に次の文を書き込む。

```
<script src="matter.js"></script>
```

3.2.2 項で作成した `helloworld.js` のソースコードを次のように書き換える。
”Hello World”という文字列から”o”の字が落下したら正しい実行結果である。

```
var Engine      = Matter.Engine,
    World       = Matter.World,
    Bodies      = Matter.Bodies;

var ball, ground;

function setup()
{
    createCanvas(300, 300);
    var engine    = Engine.create();
    var world     = engine.world;
    Engine.run(engine);
    ball         = Bodies.circle(169, 118, 6);
    ground       = Bodies.rectangle(150, 300, 300, 30, {isStatic:true});
    World.add( world, ball );
    World.add( world, ground );
}

function draw()
{
    background('black');
    stroke('white');
    fill('white');
    text('Hello W rld', 120, 120);
    ellipse( ball.position.x, ball.position.y, 6);
    rectMode( CENTER );
    fill('black');
    rect( ground.position.x, ground.position.y, 300, 30);
}
```

3.2.4 留意点

本研究では、ライブラリ `MathJax`, `p5.js`, `matter.js` を使用して開発を行う。次節以降はこれらのライブラリが宣言された環境を想定して説明を行う。それに伴い、以下に示す 3 点について留意されたい。

1. 先に述べたように本研究では `matter.js` に物理演算のみを担当させ、描画には `p5.js` を使用している。Daniel Shiffman 氏による `matter.js` のチュートリアル動画ではいくつかの実験を行っているが、作業環境やファイルのサイズによっては正しく動作しないことも懸念される。
2. 次節で紹介する、作成したサイトのトップページでは 1 画面上に 2 つのキャンバスを同時に表示している。これには `p5.js` の `instance mode` を使用している。`instance mode` についての解説は[13], [14]を参考にしている。
3. スライダーとボタンを実装するにあたって `p5.dom` というインタラクション用のライブラリを使用しているが、これらの機能はライブラリを使わずとも `HTML` や `JavaScript` のみで実装できる。`p5.dom` は図 3.1 のトップページから `Libraries>>p5.dom` と進んだページよりダウンロードが可能となっている。

3.3 サイト概要

4.2 節に後述する理由で、本校の物理Ⅱで使用されるプリントに基づいたサイトを作成した．本節では実装した機能について説明する．

それぞれのページの全体図は図 3.6, 図 3.7, 図 3.8 に示す．作成したサイトはトップページ, 問題 1.7.1 解説, 問題 1.7.2 解説の全 3 ページからなる．これらは図 3.3 のようにリンク付けされ, 互いに行き来することが可能となっている．紙媒体のレイアウトを極力崩さないことを意識し, トップページにはリンクを除く一切の機能を実装していない．

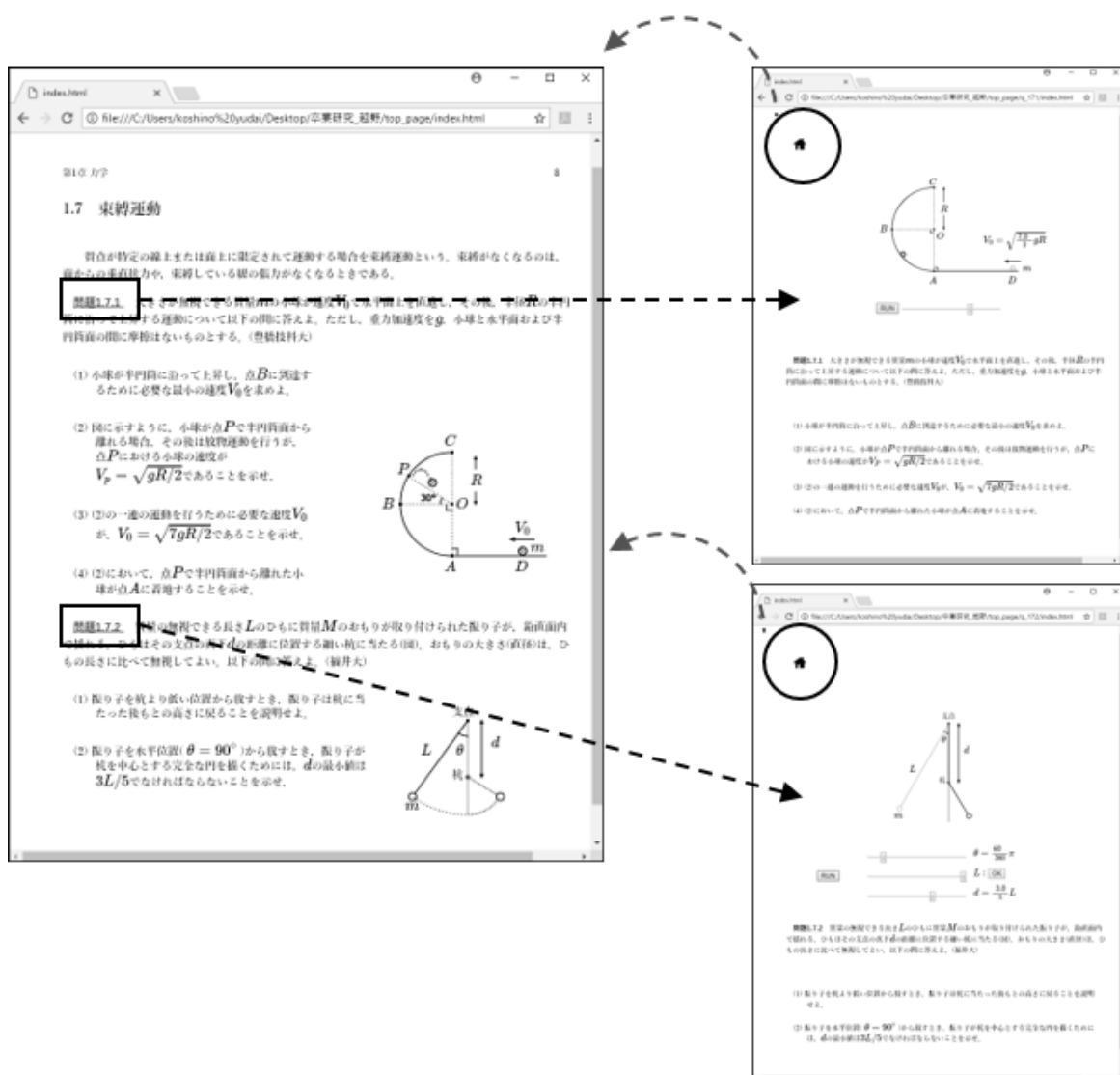


図 3.3 : ページの階層構造

それぞれの問題の解説ページに移動すると、問図、問題文、スライドバー、実行ボタンが表示される。問図はユーザがパラメータの値を変更して実行ボタンを押すと運動を始めるようになっている。図 3.4 は半円筒面上の束縛運動(a)と振り子の運動(b)を実際に動作させたときの画面である。パラメータを新しく設定して再生したい場合はページ全体をリロードする。

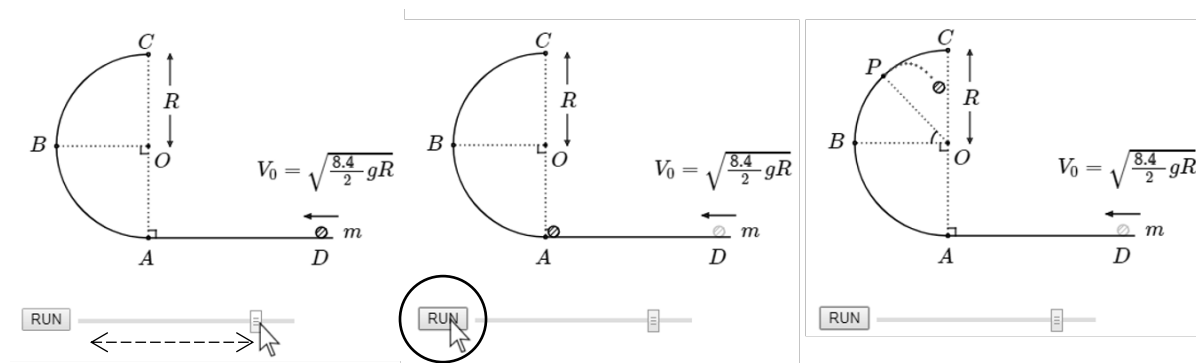


図 3.4(a)：半円筒面上の束縛運動

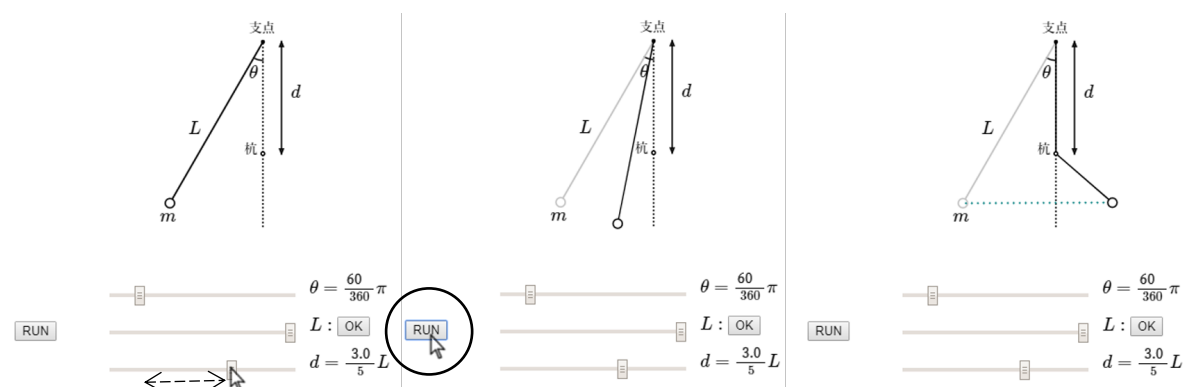


図 3.4(b)：振り子の運動

マウスを問題文の上に移動させると背景色が変わり、クリックするとその問題の解説文が展開される(図 3.5).

問題1.7.1 大きさが無視できる質量 m の小球が速度 V_0 で水平面上を直進し、その後、半径 R の半円筒に沿って上昇する運動について以下の問に答えよ。ただし、重力加速度を g 、小球と水平面および半円筒面の間に摩擦はないものとする。(豊橋技科大)

(1) 小球が半円筒に沿って上昇し、点 B に到達するために必要な最小の速度 V_0 を求めよ。

(2) 図に示すように、小球が点 P で半円筒面から離れる場合、その後は放物運動を行うが、点 P における小球の速度が $V_P = \sqrt{gR/2}$ であることを示せ。

(3) (2)の一連の運動を行うために必要な速度 V_0 が、 $V_0 = \sqrt{7gR/2}$ であることを示せ。

(4) (2)において、点 P で半円筒面から離れた小球が点 A に着地することを示せ。

問題1.7.1 大きさが無視できる質量 m の小球が速度 V_0 で水平面上を直進し、その後、半径 R の半円筒に沿って上昇する運動について以下の問に答えよ。ただし、重力加速度を g 、小球と水平面および半円筒面の間に摩擦はないものとする。(豊橋技科大)

(1) 小球が半円筒に沿って上昇し、点 B に到達するために必要な最小の速度 V_0 を求めよ。

①点 D で発射されてから A に到達するまでの質点の運動方程式は $m\vec{a} = m\vec{g} + \vec{N}$ であり垂直抗力が重力と釣り合っているため、合力は $\vec{0}$ である。
 ②半円筒面上を運動している間働いている力は保存力 $m\vec{g}$ と、進行方向に対して常に垂直な力 \vec{N} である。
 ③面を離れてからは保存力である $m\vec{g}$ のみが働く。
 これらの理由からこの問題では、着地する瞬間まで力学的エネルギー保存則が成り立つ。

力学的エネルギー保存則より、

$$\begin{aligned}\frac{1}{2}mV_0^2 &= mgR \\ V_0^2 &= 2gR \\ V_0 &= \sqrt{2gR} \quad (\because 0 < V_0)\end{aligned}$$

(2) 図に示すように、小球が点 P で半円筒面から離れる場合、その後は放物運動を行うが、点 P における小球の速度が $V_P = \sqrt{gR/2}$ であることを示せ。

(3) (2)の一連の運動を行うために必要な速度 V_0 が、 $V_0 = \sqrt{7gR/2}$ であることを示せ。

(4) (2)において、点 P で半円筒面から離れた小球が点 A に着地することを示せ。

図 3.5 : 解説文の折り畳み機能

index.html
file:///C:/Users/koshino%20yudai/Desktop/卒業研究_越野/top_page/index.html

第1章 力学
8

1.7 束縛運動

質点が特定の線上または面上に限定されて運動する場合を束縛運動という。束縛がなくなるのは、面からの垂直抗力や、束縛している紐の張力がなくなるときである。

問題1.7.1 大きさが無視できる質量 m の小球が速度 V_0 で水平面上を直進し、その後、半径 R の半円筒に沿って上昇する運動について以下の問に答えよ。ただし、重力加速度を g 、小球と水平面および半円筒面の間に摩擦はないものとする。(豊橋技科大)

- (1) 小球が半円筒に沿って上昇し、点 B に到達するために必要な最小の速度 V_0 を求めよ。
- (2) 図に示すように、小球が点 P で半円筒面から離れる場合、その後は放物運動を行うが、点 P における小球の速度が $V_P = \sqrt{gR/2}$ であることを示せ。
- (3) (2)の一連の運動を行うために必要な速度 V_0 が、 $V_0 = \sqrt{7gR/2}$ であることを示せ。
- (4) (2)において、点 P で半円筒面から離れた小球が点 A に着地することを示せ。

問題1.7.2 質量の無視できる長さ L のひもに質量 M のおもりが取り付けられた振り子が、鉛直面内で揺れる。ひもはその支点の真下 d の距離に位置する細い杭に当たる(図)。おもりの大きさ(直径)は、ひもの長さに比べて無視してよい。以下の問に答えよ。(福井大)

- (1) 振り子を杭より低い位置から放すとき、振り子は杭に当たった後もとの高さに戻ることの説明せよ。
- (2) 振り子を水平位置($\theta = 90^\circ$)から放すとき、振り子が杭を中心とする完全な円を描くためには、 d の最小値は $3L/5$ でなければならないことを示せ。

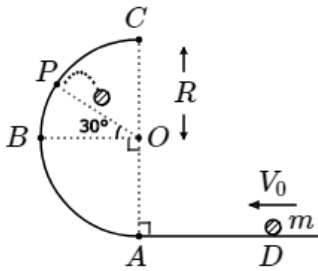
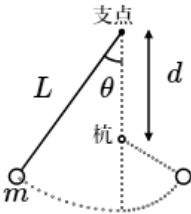
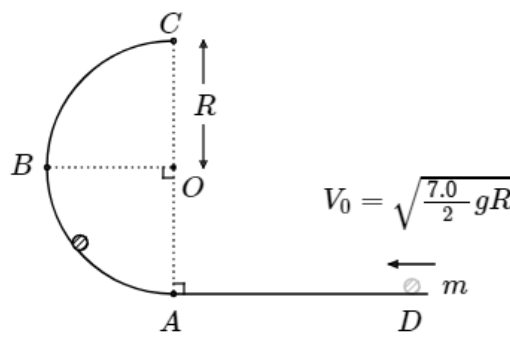



図 3.6 : トップページ

index.html

file:///C:/Users/koshino%20yudai/Desktop/卒業研究_越野/top_page/q_171/index.html

HOME



$V_0 = \sqrt{\frac{7.0}{2}gR}$

RUN

問題1.7.1 大きさが無視できる質量 m の小球が速度 V_0 で水平面上を直進し、その後、半径 R の半円筒に沿って上昇する運動について以下の問に答えよ。ただし、重力加速度を g 、小球と水平面および半円筒面の間に摩擦はないものとする。（豊橋技科大）

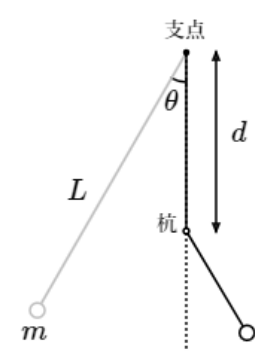
- (1) 小球が半円筒に沿って上昇し、点 B に到達するために必要な最小の速度 V_0 を求めよ。
- (2) 図に示すように、小球が点 P で半円筒面から離れる場合、その後は放物運動を行うが、点 P における小球の速度が $V_P = \sqrt{gR/2}$ であることを示せ。
- (3) (2)の一連の運動を行うために必要な速度 V_0 が、 $V_0 = \sqrt{7gR/2}$ であることを示せ。
- (4) (2)において、点 P で半円筒面から離れた小球が点 A に着地することを示せ。

図 3.7 : 問題 1.7.1 解説ページ

index.html

file:///C:/Users/koshino%20yudai/Desktop/卒業研究_越野/top_page/q_172/index.html

🏠



支点

θ

L

杭

m

d

RUN

$\theta = \frac{60}{360} \pi$

L : OK

$d = \frac{3.0}{5} L$

問題1.7.2 質量の無視できる長さ L のひもに質量 M のおもりが取り付けられた振り子が、鉛直面内で揺れる。ひもはその支点の真下 d の距離に位置する細い杭に当たる(図)。おもりの大きさ(直径)は、ひもの長さに比べて無視してよい。以下の問に答えよ。(福井大)

(1) 振り子を杭より低い位置から放すとき、振り子は杭に当たった後もとの高さに戻ることを説明せよ。

(2) 振り子を水平位置($\theta = 90^\circ$)から放すとき、振り子が杭を中心とする完全な円を描くためには、 d の最小値は $3L/5$ でなければならないことを示せ。

図 3.8 : 問題 1.7.2 解説ページ

3.4 試験的に作成した系

研究の一環として北海道大学工学部の過去試験問題の解答作成を行った。それに伴って、アンケートに使用したもの以外にも試験的に作成した系がいくつか存在する。以下に動作画面と簡単な考察を述べる。

[系 1]単振動する台上で受ける慣性力(図 3.9)

単振動させた台の上に物体を置くことで物体が抗力を受け一緒に運動する。押し付けられるタイミングで物体がめり込み、台が薄い場合はそのまますり抜けてしまう。当たり判定に関する問題があると考えられる。

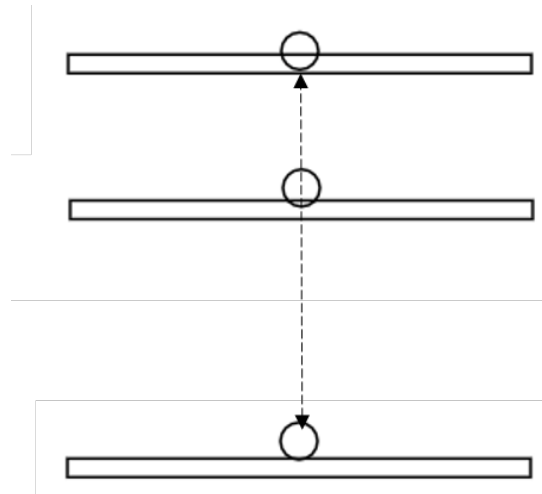


図 3.9：単振動する台上で受ける慣性力

[系 2]連成振動(図 3.10)

「2 体間の距離に応じた復元力を与える関数」としてばねを一般化することに成功したため、ばねを使ったいくつかの系が実装可能となった。連成振動は作成した全ての系の中で最も誤差の影響が小さかった。図中で球の上に表示されているのは物理エンジンとは無関係に描画した真値であり、ばね定数の設定にもよるが誤差が目に見えて現れるのは十数秒～1 分程度経過した後である。

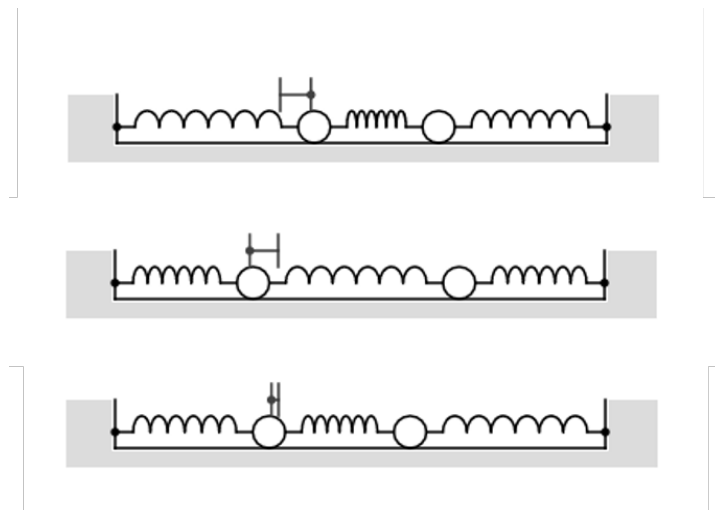


図 3.10：連成振動

[系 3]滑車とブレーキ(図 3.11)

この系を作成した時点では 3.5.3 項で述べる速度の扱いについて判明していなかったため、滑車に張力と摩擦力を与える方法でプログラムを書くと $x = r\theta$ の関係が成り立たなかった。したがってこの関係式から直接おもりを描画している。ブレーキをかけた後わずかに滑車が逆回転してしまう問題があったが理由は解明できていない。

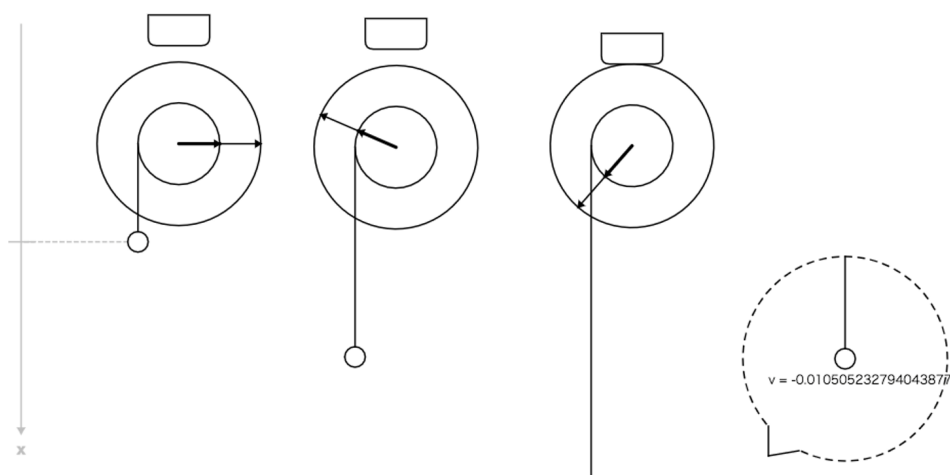


図 3.11：滑車とブレーキ

3.5 運動の描画

本節では、3.5.1 項で物理エンジンを用いる利点を述べた後、3.5.2 項ではライブラリ内で使用されている数値計算手法についてまとめる。最後に、3.5.3 項では速度の定義に関する仕様上の問題点を指摘する。

3.5.1 物理エンジンを用いる利点

運動を表現するにあたり、以下の 2 つの方法を検討した。

1. 解に基づいて軌道上に直接物体を描画する
2. 物理エンジンで近似的に軌道を計算する

精度の面では確実に前者の方が勝るが本研究では後者を選択した。本節ではその利点を説明する。

本研究では物理エンジンとして **matter.js** を採用している。**matter.js** を使用することによって質量、慣性モーメント、摩擦係数等の値は追加した物体ごとに自動的に定義され初期値が与えられる(図 3.12)。これらの値は自由に書き換えることができるため、条件の一部だけが異なる似た系を作成する際に非常に有効である。

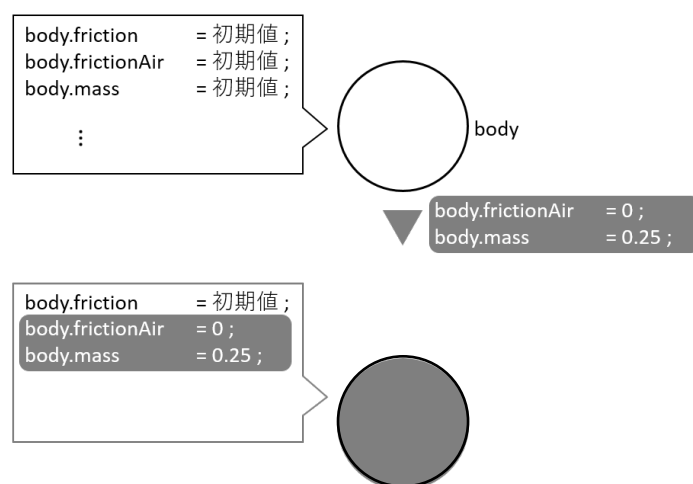


図 3.12 : 物理量代入のイメージ

具体例として自由落下の系を作成することを考える。自由落下に対する運動の解は、空気抵抗のない場合には 2 次関数の形をしている。しかし空気抵抗ありの条件に変わると解は指数関数を含んだ形になる。**matter.js** を利用しているならば両者の違いは「**frictionAir**」という値が 0 か正の数かということに限られる

ため、この値を書き換えれば良い。対して前者の方法では上記の 2 つの異なる運動の解を別々に実装する必要がある。

3.5.2 Verlet integration

`matter.js` はライブラリ内で Verlet integration[15]という数値計算手法を用いて運動方程式の解、すなわち物体の軌道を計算している。

時間幅 $\Delta t(> 0)$ を定義し、描画の開始と同時にこの時間幅で時刻をカウントする。描画の開始から n 回のインクリメントが行われた時刻 $t_n = n\Delta t$ における物体の位置を $\vec{r}(t_n) = \vec{r}_n$ とし、中央差分を用いて各時刻での加速度を次のように定義する。

$$\begin{aligned}\frac{\Delta^2 \vec{r}_n}{\Delta t^2} &= \frac{\frac{\vec{r}_{n+1} - \vec{r}_n}{\Delta t} - \frac{\vec{r}_n - \vec{r}_{n-1}}{\Delta t}}{\Delta t} \\ &= \frac{\vec{r}_{n+1} - 2\vec{r}_n + \vec{r}_{n-1}}{\Delta t^2} \equiv \vec{a}_n\end{aligned}\quad (1)$$

時刻 t_{n+1} における物体の位置はこれを変形することにより

$$\vec{r}_{n+1} = 2\vec{r}_n - \vec{r}_{n-1} + \vec{a}_n \Delta t^2 \quad (2)$$

と求まる。ここで物体に働いている力 \vec{F}_n と質量 m が定義されているならば、ニュートンの運動方程式により上式は

$$\vec{r}_{n+1} = 2\vec{r}_n - \vec{r}_{n-1} + \frac{\vec{F}_n}{m} \Delta t^2 \quad (3)$$

と書くことができる。Verlet integration では(3)式を利用して次の瞬間(時刻 t_{n+1})における物体の位置を計算している。

続けて、この計算の誤差について考察する。物体の位置 $\vec{r}(t + \Delta t)$ 、 $\vec{r}(t - \Delta t)$ を時刻 t のまわりでそれぞれテイラー展開すると

$$\vec{r}(t + \Delta t) = \vec{r}(t) + \frac{\dot{\vec{r}}(t)}{1!} \Delta t + \frac{\ddot{\vec{r}}(t)}{2!} \Delta t^2 + \frac{\vec{r}^{(3)}(t)}{3!} \Delta t^3 + O(\Delta t^4) \quad (4)$$

$$\vec{r}(t - \Delta t) = \vec{r}(t) - \frac{\dot{\vec{r}}(t)}{1!} \Delta t + \frac{\ddot{\vec{r}}(t)}{2!} \Delta t^2 - \frac{\vec{r}^{(3)}(t)}{3!} \Delta t^3 + O(\Delta t^4) \quad (5)$$

と書くことができる。ここで(4)式、(5)式の差をとり $\vec{r}(t + \Delta t)$ について変形すると次のようになる。

$$\begin{aligned}\vec{r}(t + \Delta t) + \vec{r}(t - \Delta t) &= 2\vec{r}(t) + \ddot{\vec{r}}(t) \Delta t^2 + O(\Delta t^4) \\ \vec{r}(t + \Delta t) &= 2\vec{r}(t) - \vec{r}(t - \Delta t) + \ddot{\vec{r}}(t) \Delta t^2 + O(\Delta t^4)\end{aligned}\quad (6)$$

上記の方法で導いた(6)式は $\Delta t \rightarrow 0$ の極限で正しくこの場合の真値といえる．これと(3)式を比較したとき，Verlet integration により導いた(3)式で無視されている $O(\Delta t^4)$ が誤差の項である．したがって運動の開始から時間が経つとこの誤差が蓄積され次第に目に見えた形で現れることになる(図 3.13)．

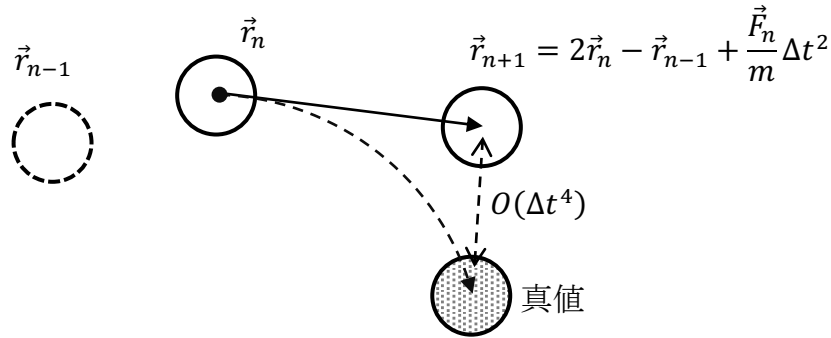


図 3.13 : Verlet integration の計算誤差

3.5.3 速度の定義に関する問題点

物体に定義されているプロパティの中に **velocity** というベクトルがある．このベクトルの大きさを計算し，**speed** という名称で再定義していることから **velocity** が速度にあたる量であることは間違いないと思われる．しかし，この値を扱う上での重大な問題点があるため以下に指摘する．

次に示すのはライブラリ **matter.js** (release 0.10.0) の 568 行目からの内容である．

```
// from the previous step
var frictionAir = 1 - body.frictionAir * timeScale * body.timeScale,
    velocityPrevX = body.position.x - body.positionPrev.x,
    velocityPrevY = body.position.y - body.positionPrev.y;

// update velocity with Verlet integration
body.velocity.x = (velocityPrevX * frictionAir * correction)
    + (body.force.x / body.mass) * deltaTimeSquared;
body.velocity.y = (velocityPrevY * frictionAir * correction)
    + (body.force.y / body.mass) * deltaTimeSquared;

body.positionPrev.x = body.position.x;
```

```
body.positionPrev.y = body.position.y;
body.position.x += body.velocity.x;
body.position.y += body.velocity.y;
```

上記のプログラム中に現れる `frictionAir`, `correction` を `console.log()`関数で確認したところ、空気抵抗なしの系を作成した場合にはこの値は 1 であることが確認できた。これを考慮して \vec{r}_{n+1} について式をまとめると

$$\vec{r}_{n+1} = \vec{r}_n + \vec{v}_n$$

となっている(ただしここで \vec{v}_n は (body.velocity.x, body.velocity.y)を表す)。つまり `matter.js` における「velocity」は物理的な意味での速度でなく

$$\vec{v}_n = \vec{r}_{n+1} - \vec{r}_n$$

の式で表される量を指す。したがって物理的な意味で正しい「速度」を得るためにはこの値を Δt で割る必要がある。

この変数 Δt を呼び出すことができなかったため、次のような実験を行い Δt がおよそ $50/3=16.666\dots$ という定数であることが分かった。(追記：本論文の編集中にこの変数を表示する方法が見つかり、以下の実験値が正しいことが確かめられた。しかしその値を変数として呼び出して計算に使うことは可能となっていない)。

[実験]

座標(0, 0)から質量の分かっている物体を落下させる。力学的エネルギー保存則から、任意の時刻における物体の速度は落下した距離 y を用いて

$$v = \sqrt{2gy}$$

と表すことができる。これを用いて Δt は次のように求まる。

$$\Delta t = \frac{\sqrt{2gy}}{\text{body.velocity.y}}$$

[結果]

初めは小数点第 3 位程度から値がぶれていたが、時間が経過するほど $50/3$ に値が近づいた。この数値の意味については不明である。

第 4 章

結論

4.1 はじめに

本章では，4.2 節で実施した評価アンケートの結果と考察を示し，続けて 4.3 節で総括と今後の課題についてまとめる．

4.2 評価アンケート

本研究は数値計算誤差の評価やプログラム言語の理論面は題材にしていないため，ここまでの研究成果に対し客観的な評価を得るためのアンケートを実施した．概要は次のとおりである．

対象者	：平成 29 年度物理Ⅱの受講者 8 名(本校 4 年生) + 平成 29 年度編入学試験の経験者 4 名(本校 5 年生)の計 12 名
評価方法	：1(まったく当てはまらない)～5(よく当てはまる)の 5 段階方式

本校で 4 年次の選択科目として開講されている物理Ⅱでは，過去の編入学試験問題を使って授業を行っている．受講者の多くが編入学を目指す学生であるため本研究の評価に協力してもらう対象として適切であると判断した．そこで物理Ⅱの授業で実際に使用されているプリントの一部を基に解説ページを作成し，これを実際に体験してもらった後でアンケートの回答をしてもらった．また，5 段階の評価とは別に自由記述式でいくつかの質問に答えてもらった．以下にアンケート項目と評価の平均，自由記述で寄せられた意見を示す．

表 4.1：アンケート結果

アンケート項目	評価の平均
教科書不足を実感した経験はあるか	4.58
問題文の意味をつかめない，運動が想像できないという理由で困った経験があるか	4.25
運動の様子を見て理解の助けになりそうか	5.00
ページのレイアウト，折り畳み式の解答は見易いか	4.67

[寄せられた意見]

- ・ 自分で座標軸を設定したい
- ・ グラフの表示機能が欲しい
- ・ 変数の値をリアルタイムに表示して欲しい
- ・ 時間の停止・巻き戻しが自由にできると良い
- ・ ページを PDF で保存したい
- ・ 画面上で運動をリロードできたほうが使いやすい
- ・ 電磁気版の開発を希望する
- ・ ばね、ひも等の組み合わせから自分で系を作成できると助かる
- ・ 色を有効に活用したほうが見やすいと思う
- ・ 解答のどの部分が運動に関わっているかをわかりやすく表示して欲しい
- ・ マウスのドラッグ機能を使って仕事の定義や 3 次元極座標の解説をしてはどうか

アンケートの結果を受けての考察を述べる。「運動の様子を見て理解の助けになりそうか」という質問に対して十分な評価が得ることができたことから本研究の目的の 1 つである、想像力の差によって力学の進度が左右される問題の軽減は達成できたと言える。しかし、画面上で運動をリロードできないことをはじめとした機能性の悪さや、見た目に関しての指摘が多く寄せられており「ページのレイアウト、折り畳み式の解答は見易いか」という質問に対しての評価が低く、課題の残る結果となった。

4.3 総括と今後の課題

本研究の成果をまとめる。

1. 動く図で運動を確認できる力学の e ラーニングサイトを開発することを目標とした
2. いくつかの系を実際に正しく動かすことに成功した
3. 現時点までに実装できた機能をまとめ、評価のアンケートを実施した

今後は、アンケートの意見を反映した新機能の実装、解説記事を充実させることを目標とし、最終的には実際に Web 上に公開することを目指す。また今回作成したサイトでは、動作させる端末やブラウザによって挙動に違いが出るという問題が発生した。原因が現時点で解明できていないため、この問題を解決することが喫緊の課題である。

謝辞

本論文の作成にあたり，終始ご指導と激励を賜りました指導教員の浅水仁教授，ならびに研究室の先輩である松井太我氏に心から感謝致します．なお，梅津裕志准教授には研究を進める中で幾度となく考察に付き合ってもらいました．特に 3.5 節に述べる内容に関してはその時のご助言がなければこうして論文に掲載することもできなかったと考えております．澤柳博文教授には貴重な授業の時間を割きアンケートの実施にご協力頂きました．同時に，お二方は私自身の大学編入学に関するご相談に乗って頂き，本研究の題材である物理を教えて下さった方々でいらっしやいます．ここにあわせて感謝の意を表します．また本研究の趣旨を理解し快くアンケートに協力して頂いた釧路工業高等専門学校の学生の皆様に感謝を申し上げます．

最後になりましたが，長くつらい受験勉強をともに乗り越えてくれた~~やかま~~
~~しい~~友人達，特に林君，野原君，江藤君，今井君の 4 人には是非ともこの場を借りてお礼を申し上げたいと思います．彼らがいなければ，一時期は口にも出さなかった「編入」をこうして研究背景にすることは決してありませんでした．心から感謝しています．

平成 30 年 2 月

釧路工業高等専門学校 電子工学科 第 44 期
越 野 雄 大

参考文献

- [1]文部科学省,“平成 29 年度学校基本調査,” http://www.mext.go.jp/component/b_menu/other/_icsFiles/afieldfile/2017/12/22/1388639_3.pdf
- [2]“EMAN の物理学,”<http://eman-physics.net/>
- [3]“物理のかぎしっぽ,”<http://hooktail.sub.jp/>
- [4]“KIT 物理ナビゲーション,”<http://w3e.kanazawa-it.ac.jp/math/physics/>
- [5]“KIT 数学ナビゲーション,”<http://w3e.kanazawa-it.ac.jp/math/>
- [6]“MathJax,”<https://www.mathjax.org/>
- [7]gilberthouse.math,“MathJax の使い方,”<http://gilbert.ninja-web.net/math/mathjax1.html>
- [8]新潟大学工学部,“MathJax の使い方,”<http://www.eng.niigata-u.ac.jp/~nomoto/download/mathjax.pdf>
- [9]“p5.js,”<https://p5js.org/reference/>
- [10]“Processing,”<https://processing.org/>
- [11]“matter.js,”<http://brm.io/matter-js/>
- [12]YouTube,“The Coding Train,”<https://www.youtube.com/user/shiffman>
- [13]“Multiple p5.js sketches in separate divs on one HTML page,”
<http://www.joemckaystudio.com/multisketches/>
- [14]YouTube,“9.11:Instance Mode(aka"namespacing")-p5.js Tutorial,”<https://www.youtube.com/watch?v=Su792jEauZg>
- [15]Saylor Avcademy,“Verlet integration,”<https://www.saylor.org/site/wp-content/uploads/2011/06/MA221-6.1.pdf>

付録

問題 1.7.1 半円筒面上の束縛運動

```
var q171 = function(_p5)
{
    //about create world
    var canvas, engine, world;
    var g      = 0.001,
        dt      = 50/3;
    //bodies
    var pointO, ball;
    //default parameters
    var R      = 80,
        size    = 5,
        V0      = Math.sqrt(7*g*R/2)*dt*0.988,
        V0_crct = V0 / dt;
    //dynamic parameter
    var x_record = [],
        y_record = [];
    var x_stop, y_stop, theta_stop;
    var Xp      = 0, Np      = 0;
    var cnt      = 0;
    var sw_r = 0;

    _p5.setup = function()
    {
        canvas      = _p5.createCanvas(400, 300);
        engine      = Engine.create();
        world      = engine.world;
        world.gravity = { x:0, y:0, scale:0 };
        _p5.noFill();
        _p5.strokeWeight(1.5);
        _p5.stroke('black');

        pointO      = new fixedPoint( _p5.width/2, _p5.height/2, 1.2, 'black' );
        ball         = new Particle( _p5.width/2 + 2*R, _p5.height/2+R, size );
        //slider
        slider = _p5.createSlider( 0, 10, 7, 0.01 );
        slider.position( _p5.width/3, _p5.height );
        slider.style('width', '200px');
        //button
        button1 = _p5.createButton('RUN');
        button1.position( _p5.width/3 -50, _p5.height );
        button1.mousePressed(run);
        function run()
        {
            {
                if(sw_r == 0 )
                {
                    Body.setVelocity( ball.body, { x:-V0, y:0 } );
                    Engine.run(engine);
                    sw_r = 1;
                }
            }
        }
    };

    _p5.draw = function()
    {
        _p5.background(255);
        var val = slider.value();
        V0      = Math.sqrt(val*g*R/2)*dt*0.988;
        V0_crct = V0 / dt;
        _p5.push();
        _p5.textFont('serif');
        _p5.fill(0);
        _p5.textSize(15);
        _p5.strokeWeight(0.5);
        _p5.text( val.toFixed(1), 370, 170 );
    }
}
```

```

_p5.pop();

//default position
_p5.push();
    //ball
    _p5.translate( _p5.width/2 + 2*R, _p5.height/2+R );
    _p5.push();
        _p5.stroke( 'silver' );
        _p5.ellipse( 0, 0, 10, 10 );
        _p5.strokeWeight(1);
        _p5.line( size, 0, 0, size );
        _p5.line( -size, 0, 0, -size );
        _p5.rotate( -_p5.PI/4 );
        _p5.line( -size, 0, size, 0 );
    _p5.pop();
    //arrow of V0
    _p5.translate( 0, -15);
    _p5.line( -15, 0, 15, 0 );
    _p5.triangle( -15,0, -10,2, -10,-2 );
_p5.pop();

//drawing half pipe
_p5.push();
    //course
    _p5.translate( _p5.width/2, _p5.height/2 );
    _p5.line( 0, R+size, (R+size)*2, R+size );
    _p5.arc( 0, 0, 2*(R+size), 2*(R+size), _p5.HALF_PI, (3/2)*_p5.PI );
    //dotted lines
    dottedLine( 0, 0, -(R+size), 0, R/2 );
    dottedLine( 0, -(R+size), 0, (R+size), R );
    //drawing point A,B,C
    _p5.ellipse( 0, R+size, 3, 3 );
    _p5.ellipse( 0, -(R+size), 3, 3 );
    _p5.ellipse( -(R+size), 0, 3, 3 );
    //right angle mark
    _p5.strokeWeight(1.2);
    var sq = 7;
    _p5.line( 0, sq, -sq, sq );
    _p5.line( -sq, sq, -sq, 0 );
    _p5.push();
        _p5.translate( 0, R+size );
        _p5.line( 0, -sq, sq, -sq );
        _p5.line( sq, -sq, sq, 0 );
    _p5.pop();
    //arrow of R
    _p5.push();
        _p5.translate( R/4, 0 );
        _p5.line( 0, 0, 0, -(R+size)/3 );
        _p5.line( 0, -(R+size)*(2/3), 0, -(R+size) );
        _p5.triangle( 0, 0, R/40, -R/20, -R/40, -R/20 );
        _p5.translate( 0, -(R+size) );
        _p5.triangle( 0, 0, R/40, R/20, -R/40, R/20 );
    _p5.pop();
_p5.pop();

//dynamic theta drawing
var Dy = pointO.body.position.y - ball.body.position.y;
var theta = Math.asin(Dy/R);
//followig
if( 0<theta && theta<Math.PI/2 && cnt<3 )
{
    _p5.push();
        _p5.translate( pointO.body.position.x, pointO.body.position.y );
        dottedLine( 0, 0, -R*Math.cos(theta), -Dy, R/2 );
        _p5.arc( 0, 0, 30, 30, Math.PI, Math.PI+theta );
    _p5.pop();
    x_stop = -R*Math.cos(theta);
    y_stop = -Dy;
    theta_stop = theta;

    var offset = 0 ;

```



```

        if( theta< Math.PI/12 ) offset=30;
        var Px = pointO.body.position.x - R*Math.cos(theta)- 30*Math.cos(theta) +offset;
        Py = pointO.body.position.y - Dy - 30*Math.sin(theta) -offset/2;
        document.getElementById("P").style.left = Px +"px";
        document.getElementById("P").style.top = Py +"px";
    }
    //after release
    if( 3<=cnt )
    {
        _p5.push();
        _p5.translate( pointO.body.position.x, pointO.body.position.y );
        dottedLine( 0, 0, x_stop, y_stop, R/2 );
        _p5.arc( 0, 0, 30, 30, Math.PI, Math.PI+theta_stop );
        //release point: P
        _p5.fill('black');
        _p5.ellipse(          x_stop-size*Math.cos(theta_stop),          y_stop-
size*Math.sin(theta_stop), 3, 3 );
        if( cnt==3 ) f();
        _p5.pop();
        //engine stop when ball collide with ground
        if( _p5.height/2+R-size < ball.body.position.y )
            World.clear(world, true);
        var y_in = Math.sqrt( R*R - (pointO.body.position.x-
ball.body.position.x)*(pointO.body.position.x) );
        if( ball.body.position.x<pointO.body.position.x && pointO.body.position.y+y_in-
size < ball.body.position.y )
            World.clear(world, true);
    }

    //ball's afterimage
    if( ball.body.position.y < _p5.height/2+R+size && 3<=cnt )
    {
        x_record.push( ball.body.position.x );
        y_record.push( ball.body.position.y );
        for( var j=0; j<y_record.length; j++ )
        {
            if( j%2 )
            {
                _p5.push();
                _p5.stroke('darkslateblue');
                _p5.ellipse( x_record[j], y_record[j], 1, 1 );
                _p5.pop();
            }
        }
    }

    var m = ball.body.mass,
        r = Vector.add( ball.body.position,
Vector.neg(pointO.body.position) ),
        e_r = Vector.normalise( r ),
        R_now = Vector.magnitude( r ),
        N = (m/R)*( Math.pow(V0_crct,2) - g*(2*R+3*Dy) ),
        NN = Vector.mult( e_r, -N ),
        W = Vector.create( 0, m*g ),
        X = ball.body.position.x - _p5.width/2;

    if( N*Np < 0 )
        cnt+=2;
    Np = N;

    if( X*Xp < 0 )
    {
        if( pointO.body.position.y < ball.body.position.y )
            cnt++;
        else
            cnt+=2;
    }
    Xp = X;

    if( cnt == 1 )
        ball.body.force = Vector.add( W, NN );
    if( cnt>=3 )

```

```

        ball.body.force = W;
    if( cnt==0 || cnt==2 )
    {
        ball.body.force.x = 0;
        ball.body.force.y = 0;
    }

    pointO.show();
    ball.show();
};

// my functions
function dottedLine(x1, y1, x2, y2, n)
{
    for( var i=0; i<n; i++)
    {
        lx = x2 - x1;
        ly = y2 - y1;
        _p5.push();
        _p5.translate( x1, y1 );
        _p5.stroke('dimgray');
        _p5.strokeWeight(1);
        _p5.fill(0);
        if(i%2)
            _p5.ellipse( lx*(i/n), ly*(i/n), 0.5 );
        _p5.pop();
    }
}

function Particle( x, y, r )
{
    var options=
    {
        friction: 0,
        frictionAir: 0
    }
    this.r =r;
    this.body = Bodies.circle(x, y, r, options);
    World.add(world, this.body);

    this.show = function()
    {
        var pos = this.body.position;
        var angle = this.body.angle;

        _p5.push();
        _p5.translate(pos.x, pos.y);
        _p5.rotate(angle);
        _p5.fill('white');
        _p5.ellipseMode( _p5.CENTER );
        _p5.ellipse(0, 0, r*2);
        _p5.push();
        _p5.strokeWeight(1);
        _p5.line( r, 0, 0, r );
        _p5.line( -r, 0, 0, -r );
        _p5.rotate( -_p5.PI/4 );
        _p5.line( -r, 0, r, 0 );
        _p5.pop();
    }
}

function fixedPoint( x, y, r, c )
{
    var options=
    {
        isStatic      :true,
        collisionFilter :0
    }
    this.body = Bodies.circle(x, y, 0.5, options);
    World.add(world, this.body);

    this.show = function()
    {

```

```

        _p5.push();
        _p5.fill( c );
        _p5.ellipseMode( _p5.CENTER );
        _p5.ellipse(x, y, 2*r);
        _p5.pop();
    }
}
function dottedArc( x, y, r, start, stop, n )
{
    for( var i=0; i<n; i++)
    {
        var delta  = stop - start;
        _p5.push();
        _p5.translate( x, y );
        _p5.fill(0);
        if(i%2)
            _p5.ellipse( r*_p5.cos( start + delta*(i/n) ), r*_p5.sin( start
+ delta*(i/n) ), 0.5 );
        _p5.pop();
    }
}
};
var myp5_1 = new p5(q171, 'q_171_img');

showBox();
var timerId;
// ボックスを表示して、タイマーを開始
function showBox() {
    document.getElementById("loading").style.display = "block"; // ボックスを表示
    timerId = setTimeout( closeBox , 5000 ); // タイマーを開始
}
// ボックスを消して、タイマーを終了
function closeBox() {
    document.getElementById("loading").style.display = "none"; // ボックスを消す
    document.getElementById("contents").style.display = "block";
    clearTimeout( timerId ); // タイマーを終了
}
}

```

問題 1.7.2 振り子の運動

```

var q172 = function(_p5)
{
    //about create world
    var canvas, engine, world;
    var g      = 0.001;
    //bodies
    var ball, p1, p2;

    var L      = 200,
        d      = 40,
        theta  = Math.PI/9,
        h1_0,
        h2_0;

    var cnt    = 0,
        Xp     = 0,
        vp     = 0,
        reached = 0;

    var EngineSwitch = 0,
        L_fixed      = 0;

    _p5.setup = function()
    {
        canvas      = _p5.createCanvas(500, 500);
        engine      = Engine.create();
        world       = engine.world;
        world.gravity = {x:0, y:0, scale:0};

        _p5.noFill();
        _p5.strokeWeight(1.5);
        _p5.stroke(0);

        p1          = new fixedPoint( _p5.width/2, _p5.height/5, 1.5, 0 );
        p2          = new fixedPoint( _p5.width/2, _p5.height/5+d, 2, 255 );
        ball        = new Particle( _p5.width/2-L*Math.sin(theta), _p5.height/5+L*Math.cos(theta),
5);

        //create slider for theta
        slider_theta = _p5.createSlider( Math.PI/9, Math.PI/2, Math.PI/9, Math.PI/360 );
        slider_theta.position( _p5.width/6, _p5.height-140);
        slider_theta.style('width', '200px');

        //create slider for L
        slider_L = _p5.createSlider( 80, 200, 200, 1 );
        slider_L.position( _p5.width/6, _p5.height-100 );
        slider_L.style('width', '200px');

        //L-fix button
        button_L = _p5.createButton('OK');
        button_L.position( _p5.width-170, _p5.height-105 );
        button_L.mousePressed(L_FIX);
        function L_FIX()
        {
            if( L_fixed == 0 )
                L_fixed = 1;
        }

    };

    _p5.draw = function()
    {
        _p5.background('white');

        if( EngineSwitch==0 )
        {
            if ( L_fixed == 0)
            {
                L = slider_L.value();
            }
        }
    }
}

```

```

theta      = slider_theta.value();
d          = (3/5)*L;
p2.body.position.y = p1.body.position.y + d;
Body.setPosition( ball.body, {x:_p5.width/2 - L*Math.sin(theta),
y:_p5.height/5 + L*Math.cos(theta)} );
}
if( L_fixed == 1 )
{
    slider_d = _p5.createSlider( (1/5)*L, (4/5)*L, (3/5)*L, (1/100)*L );
    slider_d.position( _p5.width/6, _p5.height-60 );
    slider_d.style('width', '200px');

    //start button
    button_RUN = _p5.createButton('RUN');
    button_RUN.position( _p5.width/6 -100, _p5.height-100 );
    button_RUN.mousePressed(RUN);
    function RUN()
    {
        if( EngineSwitch == 0 )
        {
            Engine.run(engine);
            EngineSwitch = 1;
        }
    }
    L_fixed++;
}
if( L_fixed >= 1 )
{
    d = slider_d.value();
    p2.body.position.y = p1.body.position.y + d;
    h1_0 = ball.body.position.y - p1.body.position.y;
    h2_0 = ball.body.position.y - p2.body.position.y;
}
}
_p5.push();
_p5.textFont('serif');
_p5.fill(0);
_p5.textSize(15);
_p5.strokeWeight(0.5);
var n_theta = theta/Math.PI*362;
_p5.text( n_theta.toFixed(0), 340, 360 );
var n_d = d/L*5;
_p5.text( n_d.toFixed(1), 345, 440 );
_p5.pop();

document.getElementById("kui").style.top = p1.body.position.y + d -30 +"px";
document.getElementById("d").style.top = p1.body.position.y + d/2 -30 +"px";
var mx = p1.body.position.x -L*_p5.sin(theta) -22*_p5.sin(theta),
    my = p1.body.position.y +L*_p5.cos(theta)-30 +22*_p5.cos(theta);
document.getElementById("m").style.left = mx +"px";
document.getElementById("m").style.top = my +"px";
var Lx = p1.body.position.x -L*_p5.sin(theta)/2 -35*Math.cos(theta),
    Ly = p1.body.position.y +L*_p5.cos(theta)/2 -35*Math.sin(theta);
document.getElementById("L").style.left = Lx +"px";
document.getElementById("L").style.top = Ly +"px";
var thx = p1.body.position.x -10*_p5.sin(theta)-20*Math.sin(theta),
    thy = p1.body.position.y +10*_p5.cos(theta);
document.getElementById("theta").style.left = thx +"px";
document.getElementById("theta").style.top = thy +"px";

var m = ball.body.mass,
    h1 = ball.body.position.y - p1.body.position.y,
    h2 = ball.body.position.y - p2.body.position.y,
    r1 = Vector.add( ball.body.position,
Vector.neg(p1.body.position) ),
    r2 = Vector.add( ball.body.position,
Vector.neg(p2.body.position) ),
    e_r1 = Vector.normalise( r1 ),
    e_r2 = Vector.normalise( r2 ),

```

```

        L1          = Vector.magnitude( r1 ),
        L2          = Vector.magnitude( r2 ),

        T1          = Vector.mult( e_r1, -(m*g/L)*(3*h1-2*h1_0) ),
        T2          = Vector.mult( e_r2, -(m*g/(L-d))*(3*h2-2*h2_0) ),
        W           = Vector.create( 0, m*g ),
        X           = ball.body.position.x - _p5.width/2;

//default drawing
_p5.push();
//default position
_p5.translate( p1.body.position.x, p1.body.position.y );
_p5.push();
    _p5.fill('white');
    _p5.arc( 0, 0, 40, Math.PI/2, Math.PI/2+theta );
    _p5.stroke('silver');
    _p5.line( 0, 0, -L*_p5.sin(theta), L*_p5.cos(theta) );
    _p5.ellipse( -L*_p5.sin(theta), L*_p5.cos(theta), 10 );
_p5.pop();
//arrow of R
_p5.push();
    _p5.translate( 20, 0 );
    _p5.line( 0, 0, 0, d );
    _p5.triangle( 0, 0, 2, 5, -2, 5 );
    _p5.translate( 0, d );
    _p5.triangle( 0, 0, 2, -5, -2, -5 );
_p5.pop();
//perpendicular line
dottedLine( 0, 0, 0, L/2 );
_p5.pop();

//counter
if( X*Xp<0 && p2.body.position.y<ball.body.position.y )
{
    if( 0<X )    cnt++;
    else          cnt--;
}
Xp          = X;

//apply force
var theta1 = Math.acos( h1/L );
var theta2 = Math.acos( h2/(L-d) );
if( cnt==0 )
{
    if( L1 < L*(80/100) || L*(150/100) < L1 )
    {
        ball.body.force = W;
        World.clear( world, true );
    }
    else
    {
        ball.body.force = Vector.add( W, T1 );
        _p5.line( p1.body.position.x, p1.body.position.y, ball.body.position.x,
ball.body.position.y);
    }
}

if( 1<=cnt )
{
    if( L2 < (L-d)*(95/100) || (L-d)*(150/100) < L2 )
    {
        ball.body.force = W;
        setTimeout( WorldClear , 200 );
    }
    else
    {
        ball.body.force = Vector.add( W, T2 );
        _p5.line( p2.body.position.x, p2.body.position.y, ball.body.position.x,
ball.body.position.y );
    }

    _p5.push();
    _p5.translate( p1.body.position.x, p1.body.position.y );

```

```

        _p5.line( 0, 0, 0, d );
        _p5.pop();
        var v = ball.body.velocity.x;
        if( v*vp<0 && p2.body.position.y<ball.body.position.y )
        {
            WorldClear();
            reached++;
        }
        vp = v;
    }

    if( reached == 1 )
    {
        _p5.push();
        _p5.stroke('teal');
        var def_x = p1.body.position.x-L*_p5.sin(theta),
            def_y = p1.body.position.y+L*_p5.cos(theta);
        dottedLine( def_x, def_y, ball.body.position.x, ball.body.position.y,
L/4 );
        _p5.pop();
    }

    ball.show();
    p1.show();
    p2.show();
};

// my functions
function dottedLine(x1, y1, x2, y2, n)
{
    for( var i=0; i<n; i++)
    {
        lx = x2 - x1;
        ly = y2 - y1;
        _p5.push();
        _p5.translate( x1, y1 );
        _p5.fill(0);
        if(i%2)
            _p5.ellipse( lx*(i/n), ly*(i/n), 0.5 );
        _p5.pop();
    }
}

function Particle( x, y, r )
{
    var options=
    {
        friction: 0,
        frictionAir: 0
    }
    this.r =r;
    this.body = Bodies.circle(x, y, r, options);
    World.add(world, this.body);

    this.show = function()
    {
        var pos = this.body.position;
        var angle = this.body.angle;

        _p5.push();
        _p5.translate(pos.x, pos.y);
        _p5.rotate(angle);
        _p5.fill('white');
        _p5.ellipseMode( _p5.CENTER );
        _p5.ellipse(0, 0, r*2);
        _p5.pop();
    }
}

```

```

function fixedPoint( x, y, r, c )
{
    var options=
    {
        isStatic      :true,
        collisionFilter :0
    }
    this.body = Bodies.circle(x, y, 0.5, options);
    World.add(world, this.body);

    this.show = function()
    {
        _p5.push();
        _p5.fill( c );
        _p5.ellipseMode( _p5.CENTER );
        _p5.ellipse( this.body.position.x, this.body.position.y, 2*r);
        _p5.pop();
    }
}

function dottedArc( x, y, r, start, stop, n )
{
    for( var i=0; i<n; i++)
    {
        var delta = stop - start;
        _p5.push();
        _p5.translate( x, y );
        _p5.fill(0);
        if(i%2)
            _p5.ellipse( r*_p5.cos( start + delta*(i/n) ), r*_p5.sin( start
+ delta*(i/n) ), 0.5 );
        _p5.pop();
    }
}

function WorldClear()
{
    World.clear( world, true );
}

};
var myp5_2 = new p5(q172, 'q_172_img');

showBox();
var timerId;
// ボックスを表示して、タイマーを開始
function showBox() {
    document.getElementById("loading").style.display = "block"; // ボックスを表示
    timerId = setTimeout( closeBox , 3000 ); // タイマーを開始
}
// ボックスを消して、タイマーを終了
function closeBox() {
    document.getElementById("loading").style.display = "none"; // ボックスを消す
    document.getElementById("contents").style.display = "block";
    clearTimeout( timerId ); // タイマーを終了
}

```