

卒業論文

ホイヘンスの原理の視覚化プログラムの開発

関西学院大学理工学部

情報科学科 西谷研究室

3518 村上 大貴

2017年3月

## 概 要

まだ未着手です。物理現象を数式だけから理解するのは非常に困難である。例えば、粒子の運動を解析する分子動力学法 (MD) では、ニュートンの運動方程式に従って粒子の運動を数値的に決定していく。しかし、数値を直接追いかけるだけでは、直感的な理解が得られない。この際、実際に粒子の運動を視覚化することによって直感的な理解を深め、学習を促進することができると考えられる。本研究では Web 上で Interactive な操作が可能で、MD の深い理解の助けとなるツールの作成を目的とする。今回は、Processing 言語を用いて、MD における粒子の運動を操作・視覚化するプログラムを作成した。また、Web 上で動作させるために、作成したプログラムを JavaScript 言語に変換した。JavaScript は動的な Web サイト構築に有効な言語の一つである。視覚表示を容易に実現する Processing 言語においては、JavaScript 言語への自動変換が組み込まれおり、Processing 言語で作成したプログラムを、ブラウザ上で動作させることができる。この自動変換には問題点があり、それらの洗い出しと解決策を検討した。プログラムを作成した結果、クラスタや凝固による粒子の振る舞いを確認することができるようになった。また、ブラウザ上で Interactive な操作が可能なため手軽に扱うことができ、今後の学習教材としても有効なものとなった。そして、ライブラリとしてプログラムの解説を行う事によって今後の継続的な発展に繋がるようにした。

# 目次

第1章	序論	2
第2章	物理学的基础知識	3
2.1	波の要素	3
2.2	波の速度と振動数と周期の関係	4
2.3	ある地点における正弦波の変位の計算方法	5
2.4	波の独立性と重ね合わせの原理	5
2.5	反射の法則	5
2.6	屈折の法則	5
2.7	回折現象	5
2.8	ホイヘンスの原理	5
第3章	開発結果	8
3.1	円形波の位相変位を視覚化したプログラム	8
3.1.1	draw モード	9
3.1.2	check モード	12
3.2	波の回折現象を視覚化したプログラム	13
3.3	波の反射を視覚化したプログラム	13
3.4	波の屈折を視覚化したプログラム	15
第4章	プログラムの制作過程と解説	16
4.1	平面波の描写	16
4.1.1	波源から波の代わりとなる円を描写するプログラム	16
4.1.2	平面波描写のアルゴリズム	17
4.1.3	描画領域のピクセル数を擬似的に変更する方法	18

4.1.4	点源とは異なる座標の点における変位の計算の実装 . . . . .	21
4.1.5	波の色の表現方法 . . . . .	21
第 5 章 総括		23

# 第1章 序論

近年, 教育に ICT (Information Communication Technology) を活用する動きが活発である. (ICT の説明をどこかでしておく) ICT 活用授業による学力向上に関する調査では, ICT を活用した授業を行った教員の 97.3% が授業における ICT 活用が児童生徒の学力向上に効果があることを認めているデータが得られている [1]. ICT 活用授業を行うための道具の 1 つとしてタブレット端末が挙げられるが, 政府は 2020 年を目標に中学および高等学校の全生徒がタブレットを携行するという計画を立てている [2]. タブレット端末を学習に活用することが出来れば, 従来指導が難しいとされていた教科も分かり易く指導を行える可能性が生まれる. そのような状況でタブレット端末の特性を活かした学習コンテンツの作成が喫緊の課題であると考ええる.

タブレット端末の利点としてインタラクティブな教材を使用できる点が挙げられるが, このような教材が効力を発揮する教科の一つに物理がある. 物理現象は, 文章や数式による記述では非常に理解しづらいものが存在する. 例えば, 高校物理で学習する干渉, 回折, 反射, 屈折といった波の性質は上記のような物理現象に相当すると考える. これを直観的に理解させ, 学習を促進する手段として, 物理現象の視覚化が適していると考ええる.

本研究では, 反射, 屈折, 回折といった波の性質の理解を助けるプログラムの作成を目的とする. プログラムはインタラクティブソフトウェアやビジュアルプレゼンテーションの作成が容易な Processing 言語により記述する.

## 第2章 物理学的基础知識

### 2.1 波の要素

ある点での振動が他の点へと伝わっていく現象を波という．波を伝える物質を媒質といい、振動によって最初に波が起きた点を波源という．図 2.1 は最も基本的な波である正弦波である．波形の中で最も高い所を山、最も低い所を谷という．隣り合う山同士、谷同士の間の距離を波長  $\lambda$  [m] という．波源から山の高さ、もしくは谷の深さを波の振幅という．

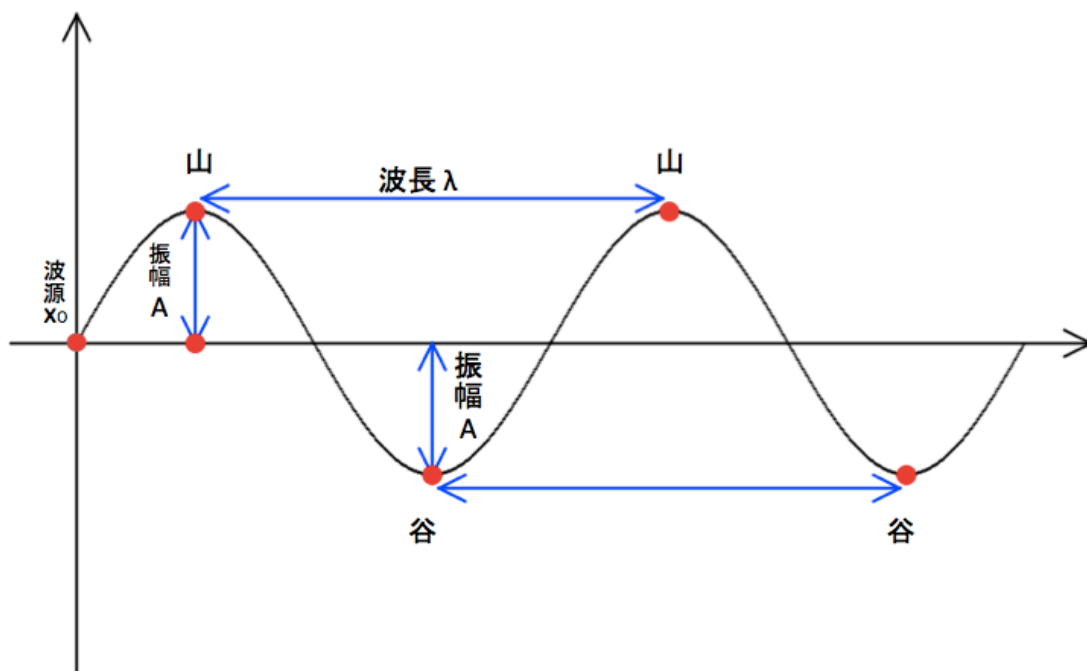


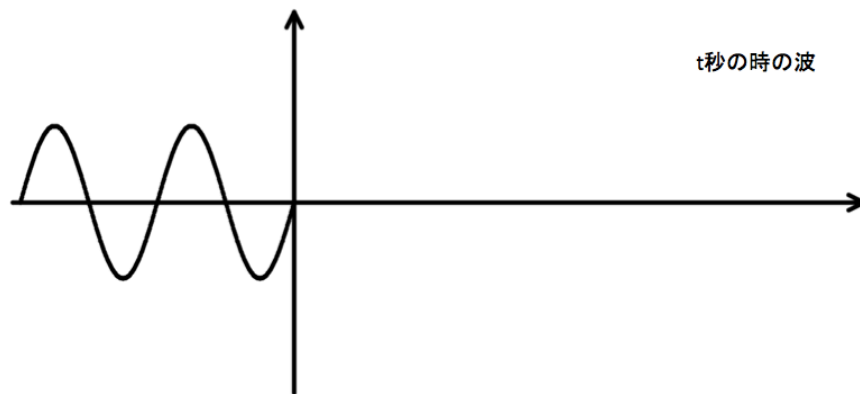
図 2.1: 波の振幅, 波長を示した図.

## 2.2 波の速度と振動数と周期の関係

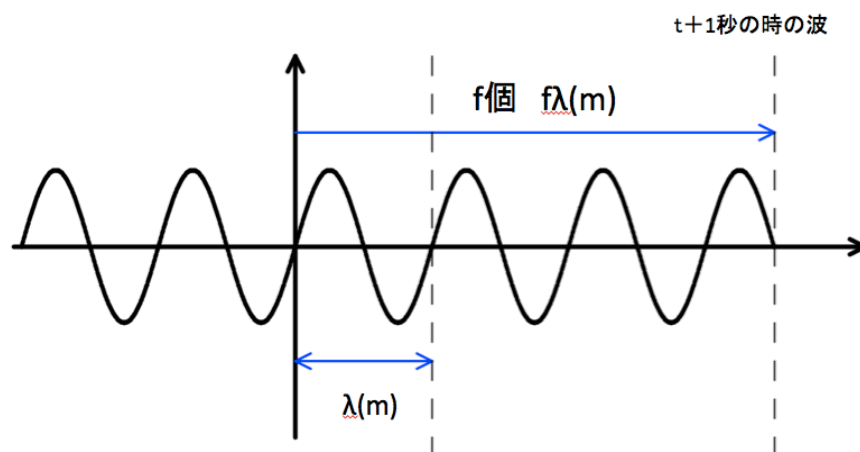
1 波長分の波が 1 秒間に発生する回数を振動数  $f[m]$  という. 波の速さ  $v[m]$  は式 (2.1) と表すことができる.

$$v = f \lambda \quad (2.1)$$

図 2.2 は波の速さ  $v$  と振動数  $f$  の関係を示した図である.



(a)  $t$  秒の時の波.



(b)  $t+1$  秒の時の波.

図 2.2: 波の速さ  $v$ , 振動数  $f$  の関係を示した図.

また 1 波長分の波が発生するまでに要する時間を周期  $T$  という。周期  $T$  と振動数  $f$  には式 (2.2) の関係が成り立つ。

$$T = \frac{1}{f} \quad (2.2)$$

## 2.3 ある地点における正弦波の変位の計算方法

ある地点における正弦波の変位を計算する際には、以下に挙げる要素が必要である。

1. 波源からの距離
2. 波長
3. 現在の時間と波源が生成された時間との差

全ての要素を同時に考慮することは難しいので、1-2 の要素と 3 の要素を分けた上で正弦波の変位の計算方法を説明する。

図 2.3(a) は、ある地点 a の変位  $y_a$  と、波源から地点 a までの距離を表したものである。波源から目標地点 (点 a) までの距離を  $d_1$  とする。図 2.3(b) は 2.3(a) の距離  $d_1$  を波長  $\lambda$  で割った余りを距離  $d_2$  とし、波源から距離  $d_2$  分離れた場所を地点 b として表したものである。

## 2.4 波の独立性と重ね合わせの原理

教科書 P188 参考書 P104 の手法で説明する。



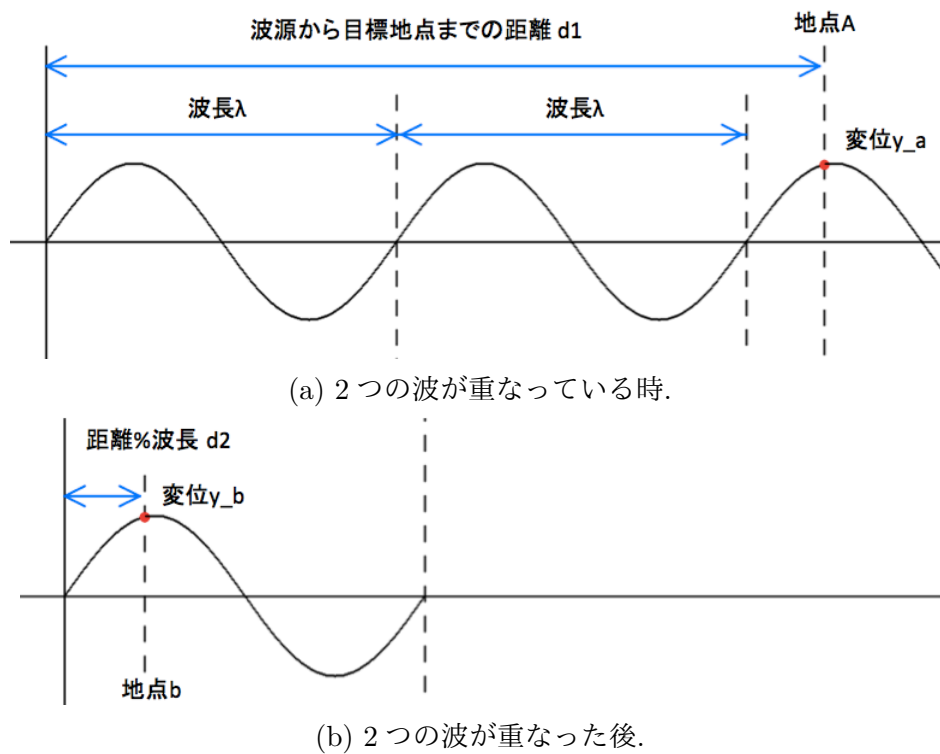


図 2.3: 波源からの距離と波長の値から、ある地点での波の変位を計算する方法.

## 2.5 反射の法則

## 2.6 屈折の法則

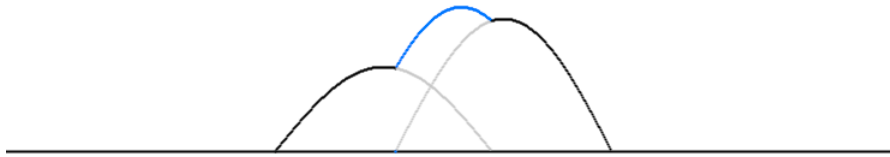
## 2.7 回折現象

## 2.8 ホイヘンスの原理

上の3つの現象をホイヘンスの原理によって説明する方法を示す.



(a) 2つの波が重なる前.



(b) 2つの波が重なっている時.



(c) 2つの波が重なった後.

図 2.4: 波の独立性と重ねあわせの原理.

## 第3章 開発結果

本章では，波の各性質をプログラムによって視覚化した実行結果を記述する．いずれのプログラムも JavaScript へ変換可能であり，Web ブラウザ上で動作させることが可能である．

### 3.1 円形波の位相変位を視覚化したプログラム

図 3.1 は指定した複数の点源から生成される円形波の位相変位をシミュレーションし，視覚化を行うプログラムである．このプログラムには draw モードと check モードという 2 つのモードが実装されており，draw モードから check モードに移行するには C キー (check の頭文字)．check モードから draw モードに移行するときは D キー (draw の頭文字) を押せばよい．

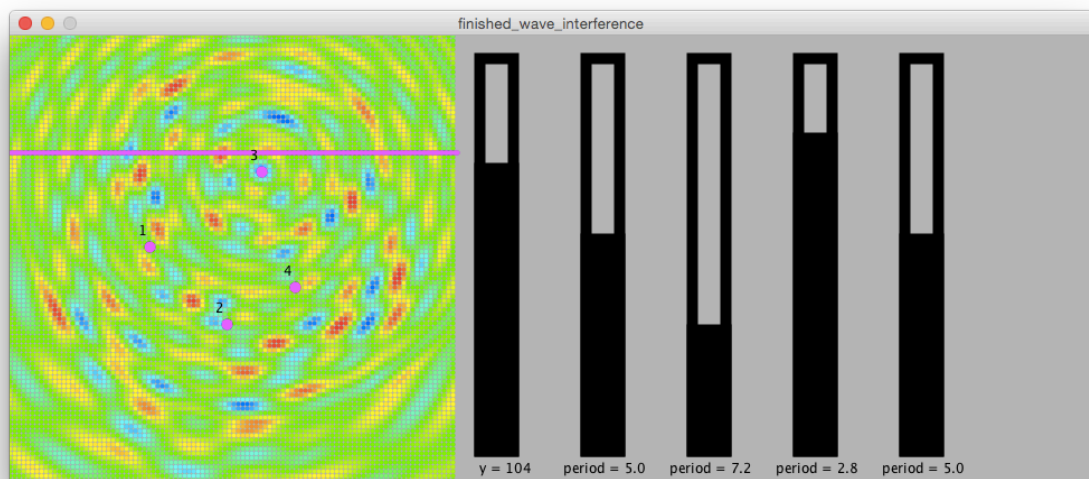


図 3.1: 波の位相変位を視覚化したプログラムの画面．

### 3.1.1 draw モード

このモードは波源から生じる円形波を描写するモードである。プログラム起動時の画面が図 3.2 である。

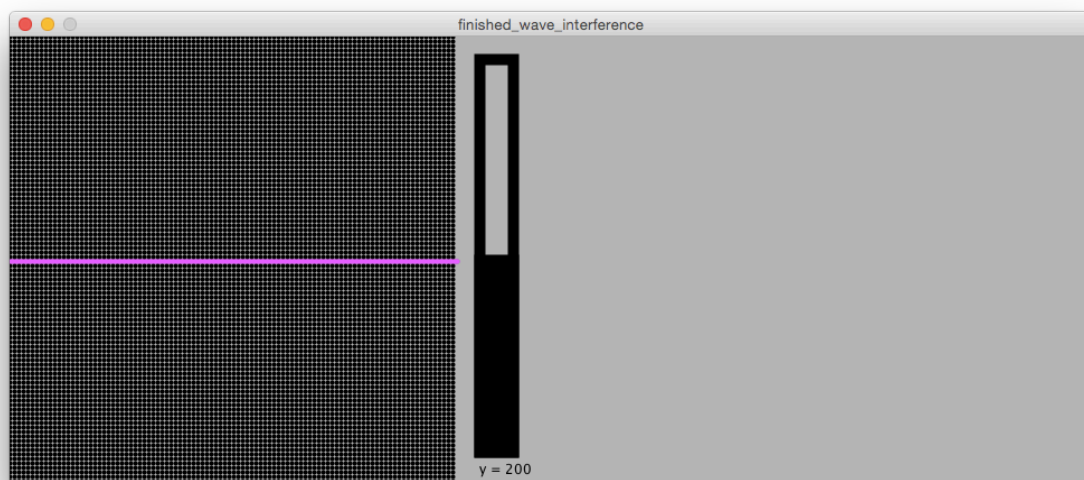


図 3.2: プログラム起動時の画面。

画面左側の黒い領域が波を描写する領域、画面右側には check モードで確認する  $y$  座標の位置を操作できるスライダーが配置されている (いきなりスライダーと言っているのか?). 図 3.2 の状態で黒い領域上のいずれかの場所をクリックすると、クリックされた座標に波源が生成される。波源が生成されたあと、画面右側にはその波源の周期を変更できるスライダーが生成される。

図 3.3 は 1 つの波源を生成した後、周期をスライダーによって変更した様子である。

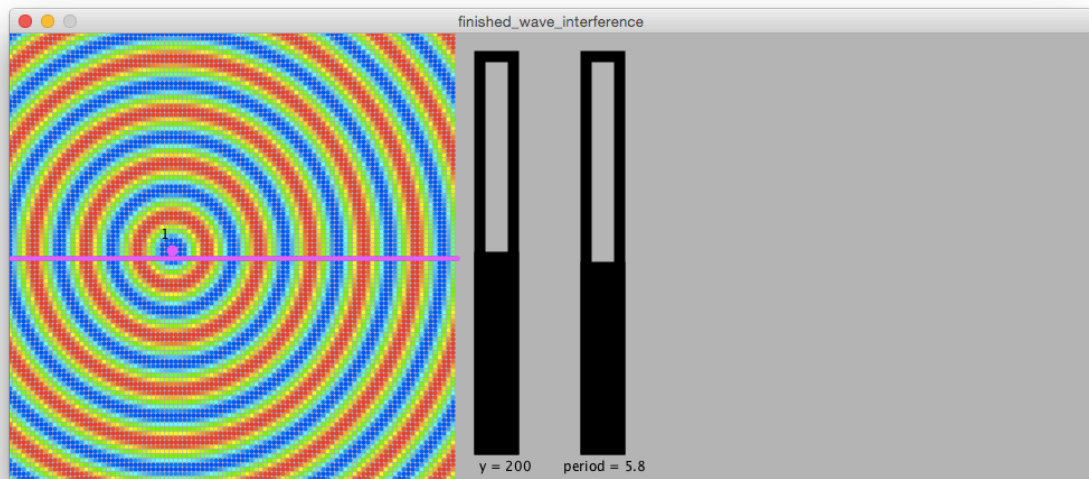


図 3.3: 1つの波の周期をスライダーで変更した画面.

スライダーが周期を変更できる状態でLキー (lambdaの頭文字) を押すと, 図3.4のように波の波長を変更できるスライダーに変化する. 周期を変更するスライダーに戻したい場合はPキー (periodの頭文字) を押せばよい.

波源は図3.5のように5個まで生成できる.

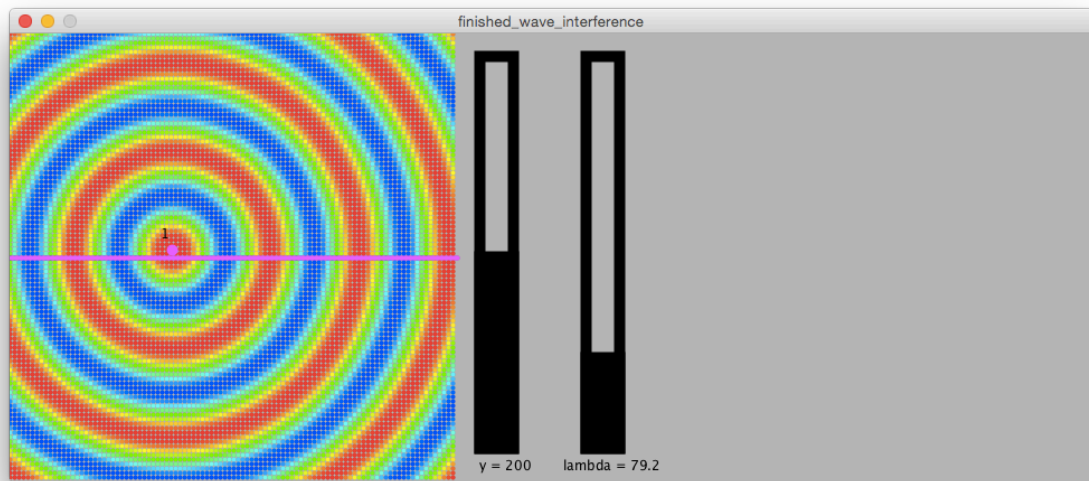


図 3.4: 波長を変更できるスライダーに変化させた時の画面.

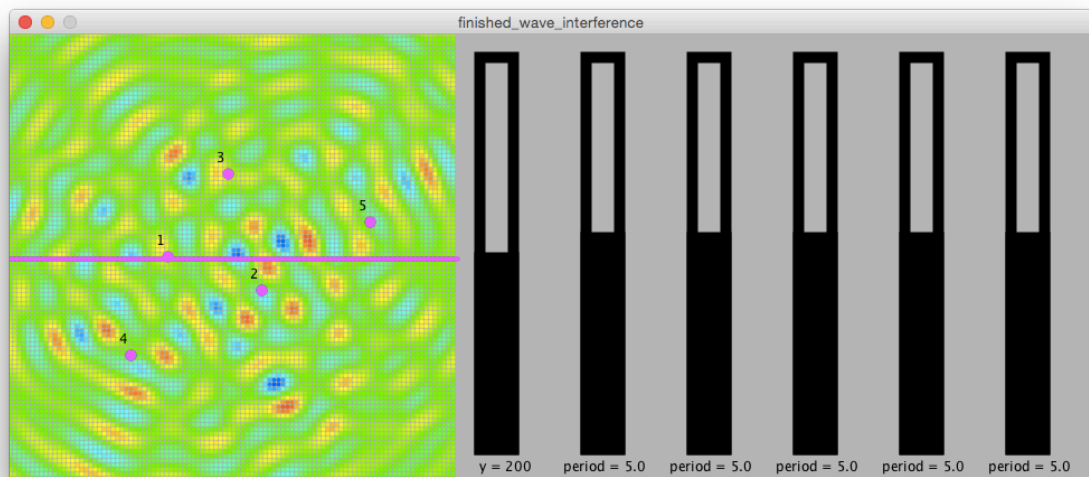
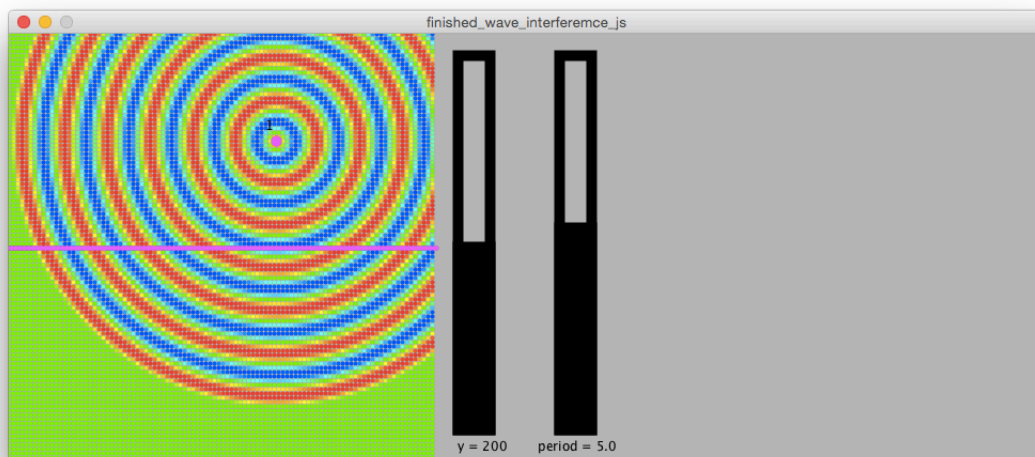


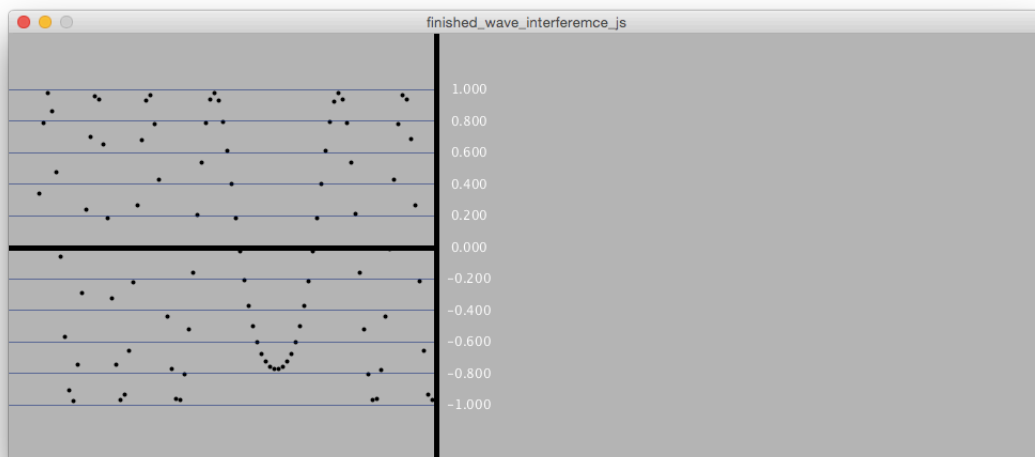
図 3.5: 波源を 5 個生成した時の画面.

### 3.1.2 check モード

このモードは draw モードに描写されている赤い線上の位相変位をリアルタイムに視覚化するモードである. 同時刻, 同座標で周期, 波長が同一な波を生成し, 波を生成して 360 フレーム目の状態を draw モード, check モードでそれぞれ描写したのが図 3.6(a),(b) である.



(a) draw モード.



(b) check モード.

図 3.6: 周期 5.0, 波長 40.0 の波が生成されてから 360 フレーム目の draw モード, check モードの画面.

## 3.2 波の回折現象を視覚化したプログラム

図 3.7 はある一定の間隔で配置された 6 つの波源から生成される円形波によって、波の回折現象の視覚化を行うプログラムである。

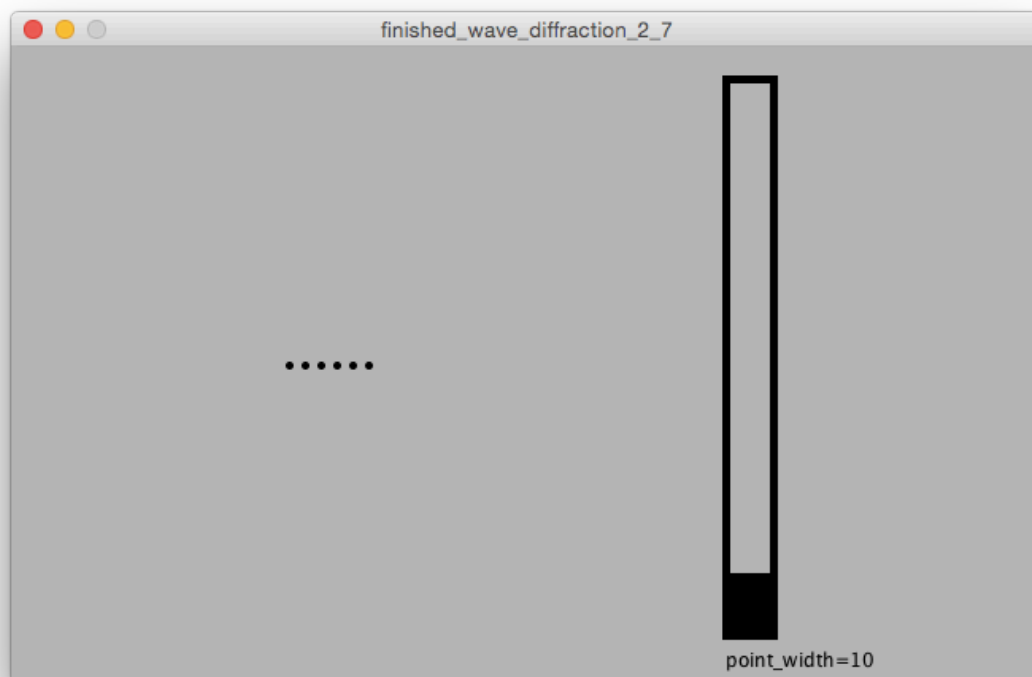


図 3.7: 回折現象視覚化プログラムの起動画面。

プログラムを起動すると図 3.7 の画面になる。この画面では画面右にあるスライダーで波源の間隔の値を調節できる。波源の間隔の値はスライダー下部に表示されている。波源の間隔の値に応じて画面左の 6 つの点が移動する。この時の点が波を生成する際の波源の間隔となる。図 3.7 の画面で S キー (Start の頭文字) を押すと、図 3.8 の画面へと切り替わる。

図 3.8 の画面は画面下部から平行に進行してきた入射波が波源に到達すると、図 3.9 のように円形波が生成され、波長に応じた挙動を描写する。

## 3.3 波の反射を視覚化したプログラム

出来なかった理由はプログラム解説の章で解説していいのか？



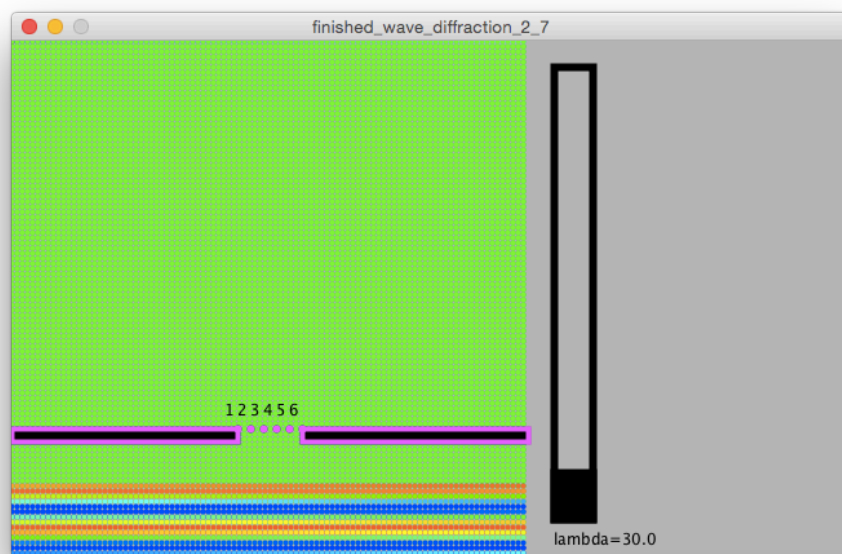


図 3.8: 入射波.



図 3.9: 回折現象.

### 3.4 波の屈折を視覚化したプログラム

## 第4章 プログラムの制作過程と解説

本章では、今回作成したプログラムのそれぞれの処理を解説する。

### 4.1 平面波の描写

回折反射屈折の性質を可視化するためには多数の波を生成し、それらに重ね合わせの原理を適用しなければならない。そこで複数の点源から波を生成した上で、重ね合わせの原理によって生じた位相変化を視覚化するプログラムを作成した。

#### 4.1.1 波源から波の代わりとなる円を描写するプログラム

最初はこの実現するために、図 4.1 のように、画面左上から右へ進行する斜めの線を入射波に見立て、入射波が波源として設定した座標を通過すると、波源から波の代わりとなる円を描写するプログラムを作成した。この後、円が重なった部分の色を変化させたり複数の円の包絡線を太く描写することを考えたが、計算アルゴリズムが非常に複雑になることや、処理速度が追いつかないことからこの方法は断念した。

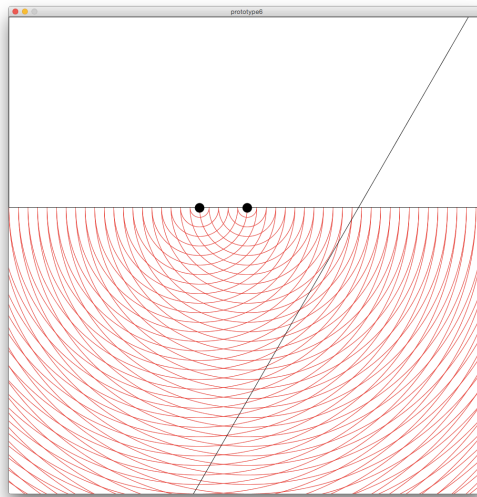


図 4.1: 波源から波に見立てた円を描写するプログラムの動作画面.

そこで各ピクセルが波源からの距離, 波源の生成された時間, 波源から生成される波の波長と周期を基にして自らの地点の位相を計算する手法を考案した.

#### 4.1.2 平面波描写のアルゴリズム

アルゴリズムは以下の手順である.

1. 平面波の描写領域の変位計算を何ピクセルごとに行うかを指定する. これにより描写領域のピクセル数を擬似的に変更する. これ以降, 説明のために擬似的に変更したピクセルの単位を擬似ピクセルと記す.
2. 描写領域がクリックされると波源の座標情報が追加される. この際 1 つの波源ごとに, 各擬似ピクセルとの距離を計算し, 各擬似ピクセルに対応した配列 (`point_distance[i][j]`) に格納している.
3. 点源とは異なる座標の点における位相の計算を全ての点源と全ての擬似ピクセルに對し行い, これによって得られた位相の値をその地点での変位に変換し, 各擬似ピクセルごとに設けた配列 (`point_[i][j]`) に足し合わせていく.
4. 各擬似ピクセルごとの変位の値を基にして, 値に応じた色の点を描写する.

5. 描画領域がクリックされた瞬間のみ手順 2 を, そうでなければ手順 3-4 を毎フレームごとに繰り返して平面波の挙動を継続的に描写し続ける.

なお, 手順 2 の段階で配列 (`point_distance[i][j]`) に値を格納しているのは, 手順 3 での異なる座標の点における位相の計算を行う際, 点源と擬似ピクセルの距離が必要であるため, あらかじめ計算させることで処理を軽くするためである.

### 4.1.3 描画領域のピクセル数を擬似的に変更する方法

4.1.2 に描写領域のピクセル数を擬似的に変更するとあるが, これを行う事により平面波描写の処理を大幅に軽減することができる. 処理の方法を以下に示す.

まず, 図 4.2 のように  $8 \times 8$  ピクセルの描写領域があるとする. これに対し 4.1.2 のアルゴリズムを適用すると, フレームごとに  $8 \times 8$  の 64 個の座標それぞれに対し位相計算を行うことになる. 一方, 図 4.3 のように  $8 \times 8$  の描写領域に対し, 変位計算を 2 ピクセルごとに行うとするならば, フレームごとの位相計算は  $4 \times 4$  の 16 個の座標に対してすればよい. つまり計算回数は図 4.2 の時と比べ, 4 分の 1 回となる.

今回作成したプログラムは描写領域を  $400 \times 400$  ピクセルとしているため, ピクセル数を変更しなければフレームごとに 160000 回の変位計算を行うことになるが, 4 ピクセルごとに変位計算を行うことで計算回数を 10000 回まで削減した.

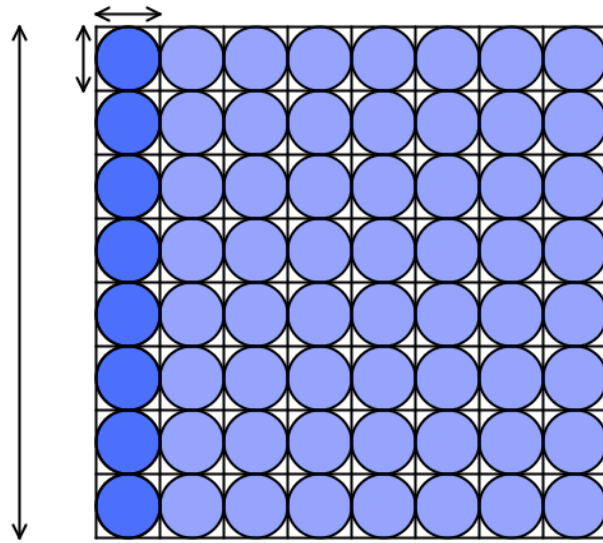


図 4.2:  $8 \times 8$  の描写領域に色を塗る場合.

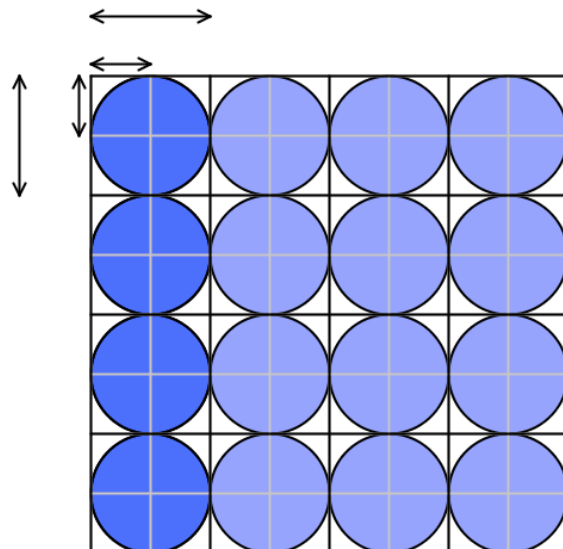


図 4.3:  $8 \times 8$  の描写領域を擬似的に  $4 \times 4$  に変更する場合.

またプログラムの擬似ピクセル数はグローバル変数の `point_regulation` の値を変えることで変更可能である。図 4.4 は擬似ピクセル数を 1(変更なし) にした時の波の描写, 図 4.5 は擬似ピクセル数を 8 にした時の波の描写である。

なお, 図 4.4, 図 4.5 以外で論文に用いているプログラムの動作画面は全て擬似ピクセル数を 4 で設定したものである。

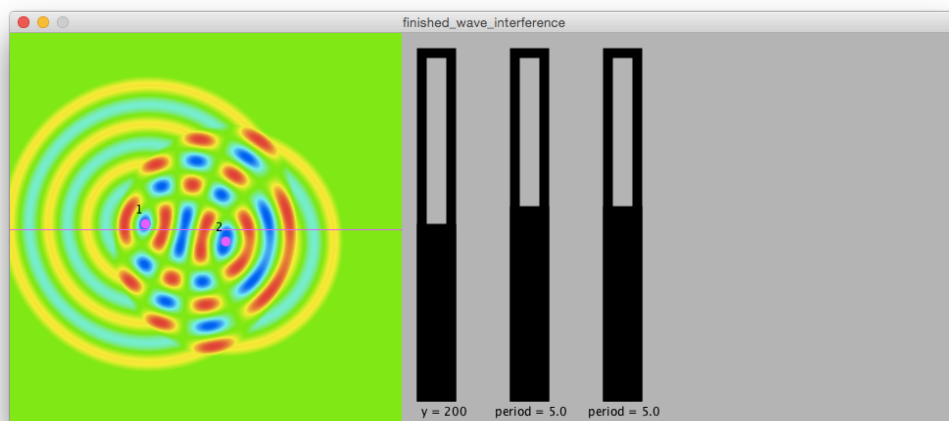


図 4.4: 擬似ピクセル数を 1 に設定した時の波の描写.

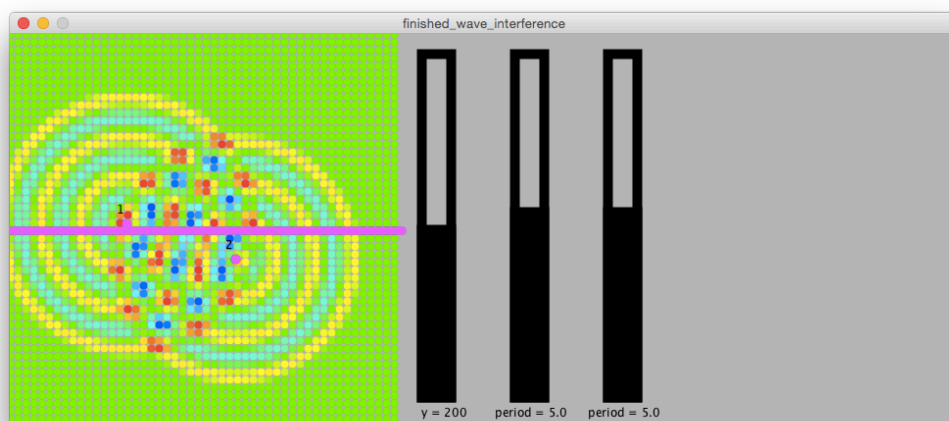


図 4.5: 画面の画素数を 8 に設定した時の波の描写.

#### 4.1.4 点源とは異なる座標の点における変位の計算の実装

物理学的背景の章で説明した (別の章の参照方法を調べる), 点源とは異なる座標の点における変位の計算を行うための Elementary\_waves クラスのメンバ関数, now\_displacement\_y は以下のように実装した.

```
float now_displacement_y(float distance, float lambda,
float period, float time_adjustment){
    float time_phase_difference;
    time_phase_difference =
        ((millis()-time_adjustment)/1000.0)*(360.0 / period);
    float a = (distance%\lambda)/lambda*(360.0);
    float y = sin(radians(-time_phase_difference+a));
    return y;
}
```

引数の distance にはある波源と, ある擬似ピクセルとの距離を代入する. lambda にはある波源から生成される波の波長, period にはある波源から生成される波の周期を代入する. time\_adjustment にはある波源が生成された際に, プログラムを起動してから経過していた時間を代入する.

phase\_difference, a(変数名はのちのち変える) が物理学的背景の章で説明したどこに対応するのかを記す.

Processing 言語にはプログラムが起動してからのミリ秒 (1/1000 秒) の数を返り値とする millis() という関数が備えられている.(ここに引用元は必要なのか?processing のリファレンス) 計算の都合上ミリ秒ではなく秒数に直した方が都合がいいので, 5 行目で millis の値を 1000 で割っている.

7 行目では sin 関数の中の値を radians() 関数によって角度の単位を「度」からラジアンに変換している.Processing のパラメータの単位はラジアンなのでこのような処理を施している.

#### 4.1.5 波の色の表現方法

擬似ピクセルごとに記録された波の変位に応じて描画する点の色を変更する.



```
stroke(point[i][j]*(230/(number_wave_point*2)),100,100,100)
```

```
point[i][j]
```

## 第5章 総括

今後の課題もここに書く．箇条書きで「」をすれば反射, 屈折を再現できると考えられる．といった風を書く．

# 謝辞

本研究を行うにあたり,終始多大なるご指導,御鞭撻をいただいた西谷滋人教授に対し,深く御礼申し上げます.また,本研究を進めるにあたり,様々な助力,知識の供給を頂きました西谷研究室の同輩,先輩方に心から感謝の意を表します.本当にありがとうございました.(修正済み)

## 参考文献

- [1] 「ICT 活用授業による学力向上に関する総合的分析評価」, 清水康敬, 日本教育工学会論文誌 32(3), (2008), 293-303.
- [2] 「教育の情報化ビジョン ～21 世紀にふさわしい学びと学校の創造を目指して～」, 文部科学省, [http://www.mext.go.jp/b\\_menu/houdou/23/04/\\_\\_icsFiles/afieldfile/2011/04/28/1305484\\_01\\_1.pdf](http://www.mext.go.jp/b_menu/houdou/23/04/__icsFiles/afieldfile/2011/04/28/1305484_01_1.pdf), p34.