

# バネモデルの シミュレータ作成

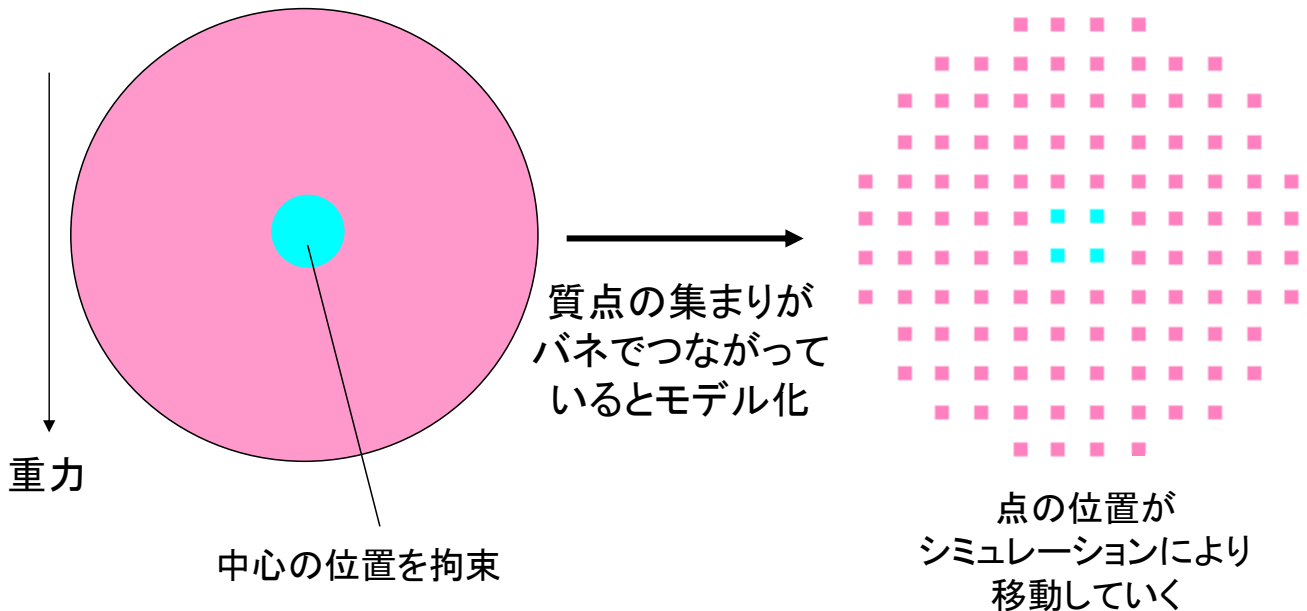
## 精密工学科プログラミング基礎 資料

1

### 目標

- バネモデルを用いて、シミュレーションをする  
– サンプルプログラムを完成・拡張させる

ゼリーのような物体を想像



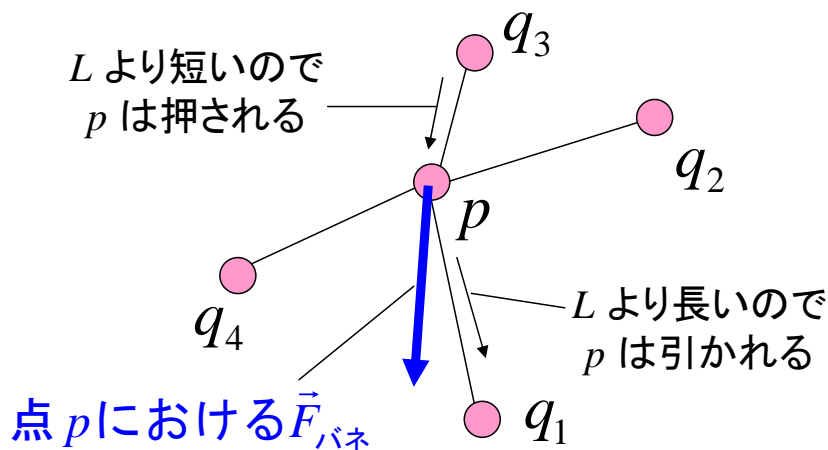
2

# バネモデル

- バネの復元力により各質点に力がかかる

$$\vec{F}_{\text{バネ}} = \sum_i k (\|\vec{q}_i - \vec{p}\| - L) \frac{\vec{q}_i - \vec{p}}{\|\vec{q}_i - \vec{p}\|}$$

$k$ :バネ係数  $L$ :バネの自然長



3

## サンプルのコンパイル方法

- OpenGL と GLUT というライブラリを使用

```
gcc -framework OpenGL -framework GLUT sample.c
```

MAC のコンパイルにおいて、  
ライブラリを指定するオプションです。

メモ:

OpenGL: 2D・3Dグラフィックス描画のためのライブラリです。

GLUT: ウィンドウを出したりマウスの操作を行うための OpenGL の補助ライブラリです。

4

# プログラムの概要

プログラム全体で使う変数 (広域変数)

- 格子点の位置と速度 → 5ページ

initPoints 関数

- 点の位置の初期化 → 5ページ

画面の更新が  
必要な時に呼ばれる

display 関数

- 絵の描画

定期的に繰り返し  
実行される

idle 関数

- 時間経過による, 点の位置・速度の更新

マウスが押されたとき  
実行される

mouse 関数

reshape 関数

- ウィンドウサイズに関わるもろもろの処理

今回は  
変更の必要なし

main 関数

- ウィンドウの初期化, GLUTへの関数の登録など

5

## データ構造

- $N \times N$  の2次元配列で位置と速度を保持

```
/* 全体で使う変数 */  
double px[N][N]; /* 点の位置の x 座標. */  
double py[N][N]; /* 点の位置の y 座標. */  
double vx[N][N]; /* 速度ベクトルの x 成分. */  
double vy[N][N]; /* 速度ベクトルの y 成分. */  
int state[N][N]; /* 状態 NON, FIX, FREE のうちどれか */
```

↑  
関数 initPoints() で、値は初期化されている  
(格子状に円盤をサンプルしている)

6

# 課題

1. 関数 `display()` で, バネを表す線分を描け  
(上下・左右の4本でつながっているとす)
  - 斜めをいれてもよい
2. 関数 `idle()` で, バネによる力を加えよ
  - ダンパをいれてもよい
3. 関数 `mouse()` で, マウスにより質点を選び  
任意の位置で固定できるようにせよ
4. バネの間をポリゴンで埋め,  
力の大きさを色で示せ

7

## 参考になる URL

- OpenGL プログラミングについて

<http://www.wakayama-u.ac.jp/~tokoi/opengl/libglut.html>

- 微分方程式の数値解法について

[http://www.akita-nct.ac.jp/~yamamoto/lecture/2003/5E/lecture\\_5E/diff\\_eq/diff\\_eq.html](http://www.akita-nct.ac.jp/~yamamoto/lecture/2003/5E/lecture_5E/diff_eq/diff_eq.html)

8