

「モデリングとシミュレーション実験」課題1

締切:2017/11/20

1 中心力場

二次元の平面 (x, y) を考えよう。この中心 $(0, 0)$ からの距離だけに依存する力が粒子に働く系を中心力場と呼ぶ。太陽の周りの衛星の周回は典型的な中心力場中の運動である。

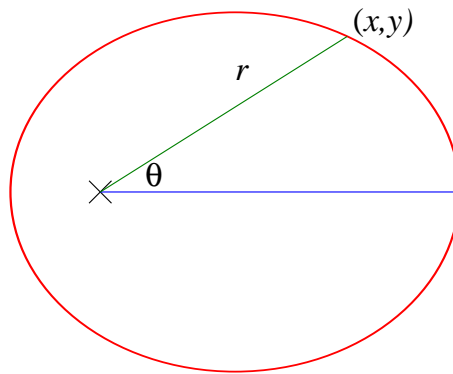


図1 極座標

粒子の位置 (x, y) に対して、

$$x = r \cos \theta, \quad y = r \sin \theta \quad (1.1)$$

によって、新しい座標 (r, θ) へ変更する。この座標系を極座標と呼ぶ。この粒子に働く力は、中心からの距離 r のみに依存する $F(r)$ であるとする。この時の運動方程式は、

$$\frac{dr}{dt} = \frac{p_r}{m} \quad (1.2)$$

$$\frac{dp_r}{dt} = \frac{p_\theta^2}{mr^3} + F(r) \quad (1.3)$$

$$\frac{d\theta}{dt} = \frac{p_\theta}{mr^2} \quad (1.4)$$

$$\frac{dp_\theta}{dt} = 0 \quad (1.5)$$

と表される。ここで p_r が r 方向の運動量、 p_θ は θ 方向の運動量（角運動量）である。式 (1.5) からわかるように、角運動量は保存する。

この運動を Runge-Kutta 法を用いて数値的に解くことを考える。ソースファイル 1 に示す `CentralForce` クラスがそれである。4 つの変数の値は $[r, p_r, \theta, p_\theta]$ の順で配列に格納ようにしている。

課題 1 コンストラクタは、質量 m と力 $F(r)$ を与えて初期化する。ソースファイル 1 中の運動方程式に相当する部分を記述しなさい。

ソースコード 1 `CentralForce.java`

```
1 public class CentralForce {
2
3     public final double m;
4     private final DoubleFunction<Double> force;
5     private final int numVar = 4;
6     private final DifferentialEquation equation;
7     private double t;
8     private double y[];
9
10    public CentralForce(double m, DoubleFunction<Double> force) {
11        this.m = m;
12        this.force = force;
13        y = new double[numVar];
14        equation = (double tt, double yy[]) -> {
15            //この部分を記述
16
17
18
19
20
21            return dy;
22        };
23    }
24
25    public void setInitial(double yy[]) {
26        System.arraycopy(yy, 0, y, 0, numVar);
27    }
28
29    public List<Point2D.Double> evolution(double tt, int nstep) {
30        double yy[][] = RungeKutta.rkdumb(y, t, tt, nstep, equation);
31        List<Point2D.Double> pList = Utils.createList();
32        for (int i = 0; i < nstep; i++) {
```

```

33         double xp = yy[0][i] * Math.cos(yy[2][i]);
34         double yp = yy[0][i] * Math.sin(yy[2][i]);
35         Point2D.Double p = new Point2D.Double(xp, yp);
36         pList.add(p);
37     }
38     t += tt;
39     for (int i = 0; i < numVar; i++) {
40         y[i] = yy[i][nstep - 1];
41     }
42     return pList;
43 }
44 }

```

このクラスには、さらに二つのメソッドがある。メソッド `setInitial(double yy[])` は、配列 `yy` で与えられる初期値を設定する。配列の中の並びに注意する。メソッド `evolution(double tt, int nstep)` は、系の状態を時間 `tt` だけ進める。ただし、その時間は `nstep` 個に分割され、小さな時間間隔で Runge-Kutta 法を実行する。結果として、粒子の x 座標と y 座標が `Point2D.Double` というクラスのインスタンスのリストとして返却される。

課題 2 クラス `CentralForce` に `main` メソッドを作成しなさい。ここで $m = 1$ 、 $F(r) = -c/r^2$ 、 $c = 1000$ とする。初期条件は

$$[r, p_r, \theta, p_\theta] = [100, 0, 0, 100]$$

とし、`tt=100`、`nstep=10000` とする。実行によって得られた (x, y) 座標の時間変化を `CentralForce.txt` へ出力しなさい。

課題 3 実行によって得られた軌道を図示する `gnuplot` スクリプトを示すとともに、図を示しなさい。

課題 4 課題 2 とは異なる初期条件についても実行し、結果を図示しなさい。

2 レポート提出

レポートは以下のように記載し、締切日の 12 時までに提出すること。

- 課題に沿って、内容を記載すること。ただし、単に課題の答えを記載するだけでは不十分である。課題の記述・理解、方法なども課題に前後して記載することで、レポートそのもので問題設定から解決までが読み取れる形式とすること。
- 「考察」として、レポートを通じて得たことを記載すること。
- 正しい日本語または英語で記述すること。Word や L^AT_EX などを使って、適切に組版すること。必要なプログラム、スクリプト、図も入れ込むこと。単なるテキストファイルの PDF は受け付けない。
- 他人のプログラム、レポートを写したと判断される場合には、0 点とする。
- レポートは PDF 形式とし、教務システムのポータルより提出すること。