6. 連成振動

2020/11/9

1　連成振動

1.1　問題設定

質量 m の n 個の粒子が、自然長 b、ばね定数 k のばねによって長さ $b(n+1)$ の空間に連結されているとします。i 番目の粒子の位置 x_i に対して、その粒子の平衡位置からのずれは

$$y_i = x_i - (i+1)b \tag{1.1}$$

です (図 1)。

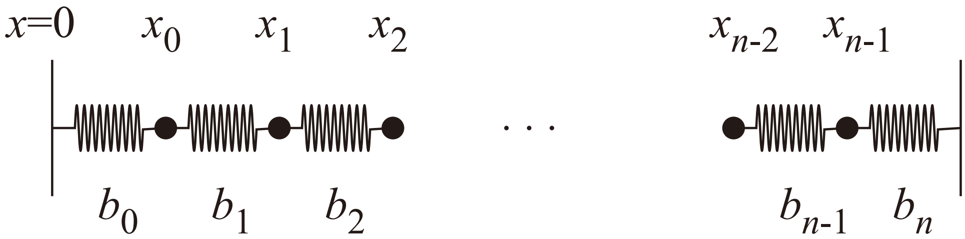

図 1　n 粒子の連成振動

二階の連立微分方程式に対して、各粒子の速度 v_i を導入し、一階連立微分方程式とします。

1.2　クラス設計

今回は、`DifferentialEquation` プロジェクト内の `coupledOscillators` パッケージを使用します。`CoupledOscillators` クラスが連成振動のクラスです。課題**??**によって、3 粒子からなる連成振動は 6 変数の一階連立微分方程式として記述することができます。例えば、6 個の変数をクラスフィールドの配列 `double y[]` に保存することにしましょう。粒子 i の変位 y_i が $2i$ 番目に、速度 v_i が $2i+1$ 番目に入ります。

$$\begin{aligned}
y_0 &\to \mathrm{y}[0]\\
v_0 &\to \mathrm{y}[1]\\
y_1 &\to \mathrm{y}[2]\\
v_1 &\to \mathrm{y}[3]\\
y_2 &\to \mathrm{y}[4]\\
v_2 &\to \mathrm{y}[5]
\end{aligned}$$

この割り当て方は、少し考えれば理解できますが、唯一の方法ではありません。従って、クラスの内部にとどめていた方が良いでしょう。

クラスの外部から、初期値を設定する、あるいは結果を取り出すには、各振動子の位置 y_i と速度 v_i が組になっているほうが理解し易くなります。そこで、調和振動子の時に使ったクラス `OscillatorStates` を各質点に割り当てます。

1.3 n 粒子

一般の n 粒子の連成振動は、行列を使って表現することができます。

$$M_{ij} = \begin{cases} 2 & i = j \\ -1 & j = i \pm 1 \\ 0 & \text{otherwise} \end{cases} \tag{1.2}$$

とすると、微分方程式は

$$m \frac{\mathrm{d}^2 y_i}{\mathrm{d}t^2} = -k \sum_{j=0}^{n-1} M_{ij} y_j \tag{1.3}$$

となります。

クラス `CoupledOscillators` のコンストラクタでは、クラス `Oscillator` の配列によって、初期値を与え、パラメタ k と b を与えることで初期化を行っています。ただし $m = 1$ です。

課題 1 クラス `CoupledOscillators` のコンストラクタの運動方程式の記述部分をソースコード 1.1 に示します。$1 \leq i \leq n-2$ に対する部分を埋めなさい。

ソースコード 1.1 運動方程式の記述

```java
        //微分方程式の記述
        equation = (double xx, double[] yy) -> {
            double dy[] = new double[2 * numOscillators];
            //0 番の粒子
            {
                int i = 0;
                int j = 2 * i;
                dy[j] = yy[j + 1];
                dy[j + 1] = -k * (2 * yy[j] - yy[j + 2]);
            }
            //1番からn-2番の粒子
            for (int i = 1; i < numOscillators - 1; i++) {
                int j = 2 * i;


            }
            //n-1番の粒子
            {
                int i = numOscillators - 1;
                int j = 2 * i;
                dy[j] = yy[j + 1];
                dy[j + 1] = -k * (-yy[j - 2] + 2 * yy[j]);
            }
            return dy;
        };
```

クラス `CoupledOscillators` の `main()` では、3 粒子の場合について、パラメタを $k = 1$、$b = 10$ とし

$$x_0 = 1,\ v_0 = 0,\ x_1 = 2,\ v_1 = 0,\ x_2 = -1,\ v_2 = 0$$

を初期値とするシミュレーションを行うように記述されています。

ソースコード 1.2　3 粒子の連成振動の様子を図示する `coupledOscillators.plt`。記号 \ は、行が見かけ上改行していても、続いていることを示している。`u` は `using` の略。

```
set terminal png fontscale 1.2
set xlabel "{/:Italic t}"
set ylabel "{/:Italic x}"
set yrange [0:40]
set xrange [0:40]
set xtic 10
set ytic 10

b=10
set output "CoupledOscillators-output.png"
plot "CoupledOscillators-output.txt" u 1:($2+b) ps 3 notitle,\
"CoupledOscillators-output.txt" u 1:($3+2*b) ps 3 notitle,\
"CoupledOscillators-output.txt" u 1:($4+3*b) ps 3 notitle
```

課題 2　実際にクラス `CoupledOscillators` を実行し、その結果を Program1.2 を使って作図しなさい。

1.4　gnuplot コマンドの補足

クラス `CoupledOscillators` の `main()` では、(t, y_0, y_1, y_2) をスペース区切りでファイルへ出力しています。gnuplot では、このようなファイルから、どの部分を使って 2 次元のグラフを描くかを指定することができます。

ソースファイル 1.2 中の `plot` の部分に注目します。`u 1:2` とは、各行の最初の数値を x に、2 番目を y としてプロットすることを表しています。つまり (t, y_0) をプロットします。次の `u 1:3` は (t, y_1) をプロットします。なお `u` は `using` の省略です。

また、`u 1:($3+5)` のように式を書くことが可能です。この場合には、$(t, y_1 + 5)$ をプロットします。

2　固有振動への分解

2.1　固有振動

講義で示したように、3 粒子それぞれの振動 $\{y_0, y_1, y_2\}$ は、三つの固有振動 $\{z_0, z_+ z_-\}$ に分けることができます。

$$z_0 = \frac{\sqrt{2}}{2}\left(y_0 - y_2\right) \tag{2.1}$$

$$z_\pm = \frac{1}{2}\left(y_0 \mp \sqrt{2} y_1 + y_2\right) \tag{2.2}$$

シミュレーション結果によって、$\{z_0, z_+ z_-\}$ が通常の調和振動であることを確かめましょう。

課題 3　`CoupledOscillators` クラスの `main()` では、ソースコード 2.1 のように、各振動子の位置を出力しています。

ソースコード 2.1　各振動子の位置を出力する部分

```
        FileIO.writeSSV(out,
                oscillators[0].t,//時刻
                oscillators[0].x,//振動子 1
                oscillators[1].x,//振動子 2
                oscillators[2].x//振動子 3
        );
```

式 (2.1) 及び (2.2) を参考に、ソースコード 2.1 の部分を変更し、固有振動 $\{z_0, z_+ z_-\}$ を出力しなさい。ソースコード 2.1 の部分を変更せず、`gnuplot` で固有振動に変換する方式でもよい。また、結果を作図し、確かめなさい。