

4. 微分方程式

2019/10/29

1 作図環境の整備

数値実験やシミュレーションの結果は、レポートや論文に図として張り込むことができなければなりません。コンピュータを自由に使うことが出来なかった時期には、データを紙に印刷し、それを手でプロットしたものを清書していました。現在では様々なデータプロットツールが利用できます。この実験では、多くの OS 上で 30 年近くも利用され続けている gnuplot (<http://www.gnuplot.info/>) を利用することにします。

1.1 gnuplot のインストール

- 配布したインストールファイルをダブルクリックし、インストールを実行する。
- 途中の「追加タスクの選択」において、「実行ファイルのディレクトリを PATH 環境変数に追加する」にチェックをするのを忘れないようにしましょう。
- 後から PATH を追加するには、「コントロールパネル」→「システムとセキュリティ」→「システム」→「システムの詳細設定」→「環境設定」、と移動して設定します。

課題 1 始めに、簡単なテストをしましょう。インストールした gnuplot を起動します。起動後、

```
plot sin(x)
```

と入力すると、正弦関数 `sin` の波形が描画されるはずです。

課題 2 gnuplot では、関数を定義し、その関数をプロットすることもできます。例えば上記の正弦関数に加えて、

$$f(x) = 2\sin(x + \pi)$$

をプロットする場合には、gnuplot を起動した後にソースコード 1.1 のようにタイプすることで、二つの関数を表示することができます。ここで `pi` は π を表します。

ソースコード 1.1 gnuplot の例

```
1 a = 2
2 f(x) = a * sin(x + pi)
3 plot sin(x), f(x)
```

2 共通のライブラリの準備

この実験では、Java を使ってモデルを実装します。その際に、ファイルへの出力や新しいリストの生成など、同じような処理を何度も行います。

課題 3 そのような共通の処理をライブラリにまとめたものをインストールしておきましょう。以下の URL より、プロジェクト MyLib をダウンロードします。

<https://github.com/modeling-and-simulation-saga/MyLib>

取得したプロジェクトをビルドすると dist ディレクトリ下に MyLib.jar が生成されます。

MyLib には以下のようなクラス、メソッドが含まれています。

myLib.rungeKutta

DifferentialEquation:微分方程式を表すインターフェイス。

メソッド double[] derivatives(double x, double y[]) を実装しなければならない。

RungeKutta:4 次の Runge Kutta 法で微分方程式を解くクラス。二つのメソッド

```
static double[] rk4(double x, double y[], double h,  
DifferentialEquation eq)
```

と

```
double[][] rk4dumb(double vstart[], double x1, double x2,  
int nstep, DifferentialEquation eq)
```

を有する。

Dynamics:具体的な微分方程式で表される系に対応する抽象クラス。

myLib.utils.FileIO:ファイル入出力用のライブラリ

```
BufferedWriter openWriter(String filename)
```

throws IOException:ファイル名を指定して writer を開く

```
void writeSSV(BufferedWriter out, Object... objects)
```

throws IOException:out へ向けて、objects で指定された対象物の列をスペース区切りで出力する

myLib.utils.Util:その他の便利なもの

```
List<T> createList():クラス T の空のリスト生成
```

```
int[] createRandomNumberList(int max)0 から max-1 までの整数のでたらめな並びの生成
```

3 Euler 法による重力中の粒子運動のシミュレーション

Euler 法の例として、重力中の粒子運動のシミュレーションを行います。その準備として GitHub からプロジェクトをダウンロードします。

<https://github.com/modeling-and-simulation-saga/DifferentialEquation>

課題 4 クラス `ThrowBallEuler.java` に、重力中の粒子運動の微分方程式

$$\frac{d^2x}{dt^2} = 0$$
$$\frac{d^2y}{dt^2} = -g$$

に対応した記述を追加しなさい。

続いて、以下の手順で実行します。

- プロジェクトウィンドウ内のクラス `ThrowBallEuler` を右クリックし、メニュー内の「クラスを実行」によって実行します。
- `$HOME\Documents\NetBeansProjects\DifferentialEquation` のしたに `ThrowBallEuler-output.txt` ができていることを確認します。

クラス `ThrowBallEuler` の `main` メソッドを見てみましょう。

```
String className = ThrowBallEuler.class.getSimpleName();
String filename = className+"-output.txt";
```

によって、クラス `ThrowBallEuler` の名前を文字列 `className` に保存しています。出力用のファイルはクラス名の後ろに `-output.txt` を追加して、命名し、以下でファイルへの出力を指定しています。

```
BufferedWriter out = FileIO.openWriter(filename);
```

最後に、結果を出力しましょう。gnuplot は、作図作業命令をファイルに保存することができます。`$HOME\Documents\NetBeansProjects\DifferentialEquation` に `throwBallEuler.plt` をソースコード 3.1 の内容で作成します。一行目が折り返されていることに注意してください。

ソースコード 3.1 の各行は以下のような作業を行います。

1. 作図を PDF で行うとともに、図の大きさや使用するフォントを指定
2. x -軸のラベルを設定
3. y -軸のラベルを設定
4. 図のタイトルを指定
5. 出力先ファイルを指定
6. `ThrowBallEuler-output.txt` からデータを読み込みプロット

ソースコード 3.1 `throwBallEuler.plt`

```
1 set terminal pdfcairo enhanced color size 29cm,20cm font "Times-New-Roman" fontscale 1.2
2 set xlabel "{/:Italic_x}"
3 set ylabel "{/:Italic_y}"
4 set title "ThrowBallEuler"
5 set output "ThrowBallEuler.pdf"
6 plot "ThrowBallEuler-output.txt" ps 2
```

NetBeans はテキストエディタとしても利用できます。ソースコード 3.1 を NetBeans を使って作成しま

しょう。プロジェクトが表示されている左のスペースにある「ファイル」というタブを開けます。対象となるフォルダで、マウス右ボタンを押し、「新規」を選びます。ファイルの種類を選ぶところから「その他」→「空のファイル」を選び、ファイルを作成します。拡張子の指定を忘れないようにしてください。あとは、マウス右ボタンで「開く」を選択することで、編集を開始することができます。

課題 5 シミュレーションの結果を作図しなさい。

このファイルをダブルクリックすると、ファイル `ThrowBallEuler-output.txt` からデータを読みだしてプロットします。出力結果は、`ThrowBallEuler.pdf` に保存されます。結果を確認しましょう。

出力ファイルが作成されない場合は、`throwBallEuler.plt` に誤りがあることが考えられます。そのような場合には、エクスプローラで、`throwBallEuler.plt` があるディレクトリに移動し、PATH 表示部分でマウスをクリックして PATH の文字列を反転させます (図 1)。その状態で `cmd` とタイプしリターンすることで、コマンドプロンプトを開きます。そこで、

```
gnuplot throwBallEuler.plt
```

とタイプすることで、エラーを含めて出力することができます。

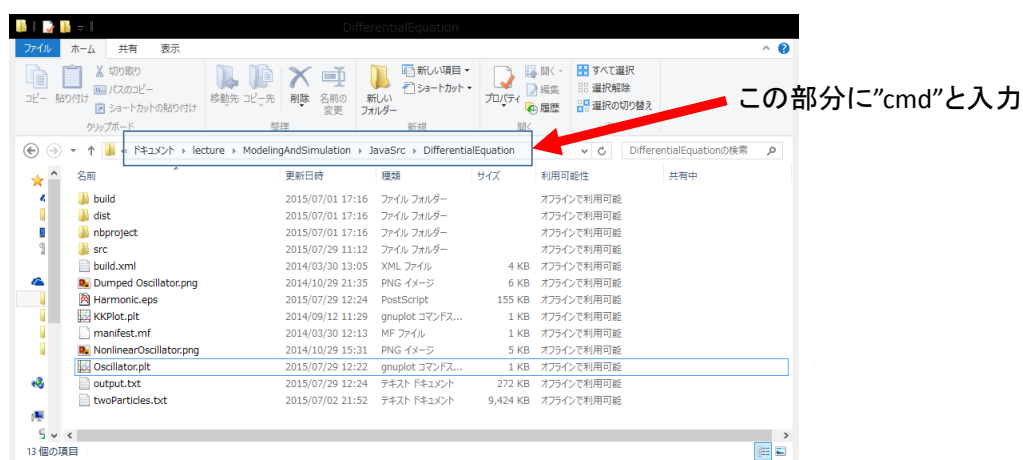


図 1 エクスプローラからコマンドプロンプトを開く。

課題 6 今回のシミュレーションでは、 x 方向及び y 方向の初速を $v_x = v_y = 10$ とし、重力加速度を $g = 9.8$ としています。従って (x, y) 面内の軌道は

$$y = \frac{v_y}{v_x}x - \frac{1}{2}g\left(\frac{x}{v_x}\right)^2$$

となります。シミュレーション結果とともに、上記の式もプロットしましょう。