

4. 微分方程式

2020/10/26

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、それを手でグラフ用紙にプロットし、さらに清書していました。現在では様々なデータプロットツールを利用することができます。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。

- 64bit 版
<https://sourceforge.net/projects/gnuplot/files/gnuplot/5.2.8/gp528-win64-mingw.exe/download>
- 32bit 版
<https://sourceforge.net/projects/gnuplot/files/gnuplot/5.2.7/gp527-win32-mingw.exe/download>

1.1 gnuplot のインストール

- ダウンロード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します。
- 日本語の使用を許可します (図 1 左)。
- 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 (図 1 右)。こうすることで、コマンドラインからも実行可能となります。
- 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。

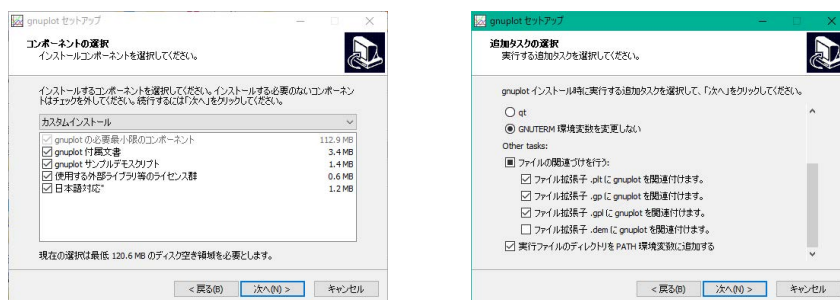


図 1 gnuplot の日本語設定とパスの設定

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 (図 2)。

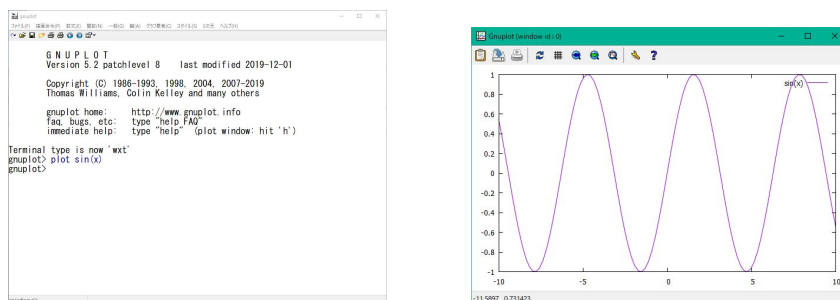


図 2 gnuplot の起動と正弦関数の描画

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 π 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 共通的ライブラリの準備

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 そのような共通的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。GitHub から、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` を生成します。

MyLib には以下のようなクラス、メソッドを含んでいます。

```
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[][] rk4mb(double vstart[], double x1, double x2,`
`int nstep, DifferentialEquation eq)`

を有する。

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。␣は、半角スペースです。_などを入力しないでください。

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。適切なテキストエディタを使って作成しましょう。図 3 に Visual Studio Code での編集画面を表示します。ファイルの保存忘れに注意してください。

1. 作図を PNG 形式で行うとともに、使用するフォントの大きさを指定
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。データのタイトルを"simulation"に設定。

ソースコード 3.1 `throwBallEuler.plt`

```
1 set terminal png fontscale 1.2
2 set xlabel "{/:Italic␣x}"
3 set ylabel "{/:Italic␣y}"
4 set title "Throw␣Ball␣(Euler␣method)"
5 set output "ThrowBallEuler.png"
6 plot "ThrowBallEuler-output.txt" ps 2 title "simulation"
7
```

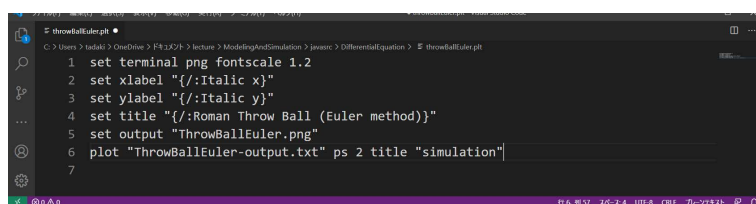


図 3 Visual Studio Code でソースコード 3.1 を編集。左上のファイル名の横にある白い○は、ファイルが保存されていないことを示している。

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 `throwBallEuler.plt` 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.png` に保存します。結果を確認しましょう。なお、出力の PNG ファイルを「ペイント」アプリケーションなどで開いていると、gnuplot からの出力できません。注意してください。

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4)。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。

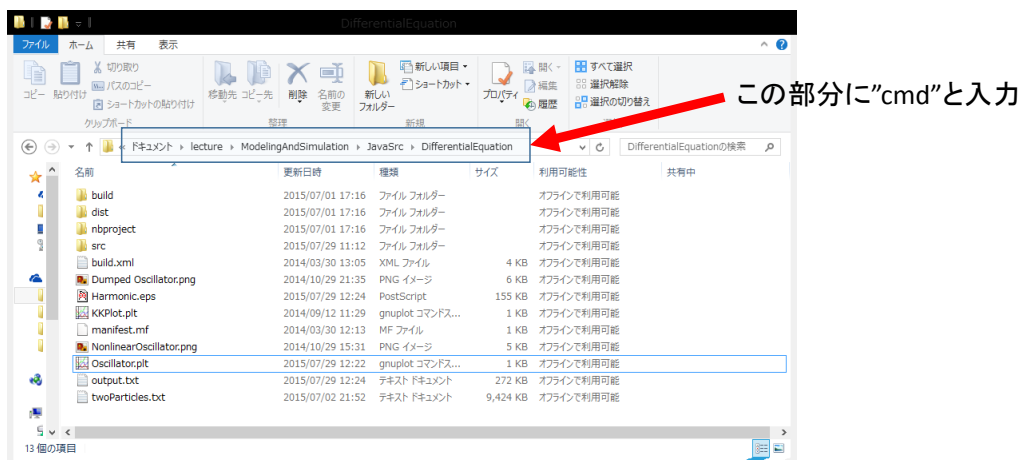


図4 エクスプローラからコマンドプロンプトを開く。

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。