2023-知能情報基礎演習 4-1

微分方程式の数値解法

2020/07

目的

　微分方程式の解をコンピュータを用いて数値的に解く方法について学習する.
今回は，直感的で分かりやすいオイラー法のアルゴリズムを C＋＋言語を用いてプログラミングする事を目的とする.

数値解法の必要性

　古くより弾道計算，構造物の振動解析，電気回路の過渡応答解析，化学反応の進行具合いの解析，近頃では宇宙ロケットの軌道計算も加わって，偏微分あるいは(常)微分方程式の初期値問題の形に定式化される実際問題は数知れないくらい多い.
　その他にも，近年の 3D(コンピュータグラフィックス)やバーチャルリアリティなどもコンピュータを使った数値計算なしでは実現不可能である.

常微分方程式の初期値問題

独立変数をt，tの未知関数(従属変数とも言う)を$x(t)$として、$x(t)$の満たすべき二つの条件，すなわち
1. 微分方程式：

$$\frac{d}{dt}x = f(x,t) \qquad (a \le t \le b) \tag{1}$$

2. 初期条件：

$$x(a) = x_0 \tag{2}$$

を与えて関数$x(t) \quad (a \le t \le b)$を求める，というのが問題である.

”連立”方程式の場合も，未知関数が $x^1(t), \cdots, x^m(t)$ と，m個あること以外は形式的には全く同じで，
1. 微分方程式：

$$\frac{d}{dt}x^i = f^i(x^1, \cdots, x^m, t) \qquad (i = 1, \cdots, m; a \le t \le b) \tag{3}$$

2. 初期条件：

$$x^i(a) = x_0^i \qquad (i = 1, \cdots, m) \tag{4}$$

を与えて $x^i(t)(a \leq t \leq b)$ を求めることが問題である.
“高階”の微分方程式，たとえば

$$\frac{d^3}{dt^3}x = f(x, \frac{dx}{dt}, \frac{d^2x}{dt^2}, t) \tag{5}$$

の様なものも，未知関数の数を増やして

$$x^1 = x, \quad x^2 = \frac{dx}{dt}, \quad x^3 = \frac{d^2x}{dt^2} \tag{6}$$

とおけば

$$\frac{dx^1}{dt} = x^2$$

$$\frac{dx^2}{dt} = x^3$$

$$\frac{dx^3}{dt} = f(x^1, x^2, x^3, t) \tag{7}$$

という 1 階微分方程式の形に帰着できる.

オイラー法

最も原始的な解法で，「刻み幅」と呼ぶ量τを定めて，独立変数のとびとびの値

$$t_n = a + n\tau \qquad (n = 0, 1, 2, \cdots) \tag{8}$$

における未知関数の値$x^i(t_n)$の近似値x_n^iを，

$$x_{n+1}^i = x_n^i + \tau f_n^i, \qquad f_n^i \equiv f^i(x_n^1, \cdots, x_n^m, t_n \tag{9}$$

によって次々と$(n = 0, 1, \cdots)$定めていく方法である．

この公式は$t = t_n$における微分方程式(3)の左辺を

$$\frac{d}{dt}x^i \fallingdotseq \frac{x_{n+1}^i - x_n^i}{t_{n+1} - t_n} = \frac{x_{n+1}^i - x_n^i}{\tau} \tag{10}$$

で置き換えたものの，分母を払って移項したものであると思えば良い．

問題:1 †

　高校生向けにオイラー法を説明する場合、どのように説明したら良いだろうか？
　オイラー法の特徴をわかりやすく説明せよ．
さらに、オイラー法の問題点があるとすればそれは何か説明せよ．

例題

それでは，簡単な例題を見てみよう．

$$\frac{d}{dt}x = -x, \qquad x(0) = 1 \tag{11}$$

この(一階の)微分方程式には解析解が存在して，

$$x(t) = e^{-t} \tag{12}$$

となる．

問題:2 †

　微分方程式(11)の解の導出過程を記述せよ．

実験：野球ボールの軌道計算

前節までで学習したオイラー法を用いて，ピッチャーが投げた野球ボールを打者が打った時のボールの軌道を計算するプログラムを C++言語を用いて作成しよう.

まず，ボールの運動に関する基礎方程式を説明する． 運動を記述するには，投射物の位置ベクトル$r(t)$と速度ベクトル$v(t)$を計算する必要がある． この運動方程式は，

$$\frac{dv}{dt} = \frac{1}{m}F_a(v) - g\hat{y}; \qquad \frac{dr}{dt} = v \tag{13}$$

となる.

ただしmは投射物の質量，$F_a(v)$は空気抵抗による力，gは重力加速度，$\hat{y}$は鉛直方向の単位ベクトルである．実際の空気抵抗の計算は非常に複雑になるので，ここでは次の近似式を用いることとする.

$$F_a = -\frac{1}{2}C_d\rho A|v|v \tag{14}$$

ここで，C_dは抵抗計数，ρ空気の密度，Aは投射物の横断面積である.

空気抵抗を無視できる場合，運動方程式は以下の方法で解析的に解くことができる.

$$r(t) = r_1 + v_1 t - \frac{1}{2}gt^2\hat{y} \tag{15}$$

ただし，$r_1 \equiv r(t=0)$と$v_1 \equiv v(t=0)$は位置と速度の初期値である.

以下にプログラムの概要をまとめる.

プログラム概要

- ボールの初期位置 r_1 および初期速度 v_1 を設定する.
- 物理パラメータ $(m, C_d$など$)$ を設定する.
- ボールが地面に着くまで，あるいは最大の刻み数になるまでループする.
  - プロット用に位置（計算値および理論値）を記録する.
  - ボールの加速度を計算する.
  - オイラー法を用いて，新しい位置 r_{n+1}および v_{n+1} を計算する.
  - ボールが地面に着いたら $(y < 0)$ ループを抜ける.
- 最大到達高さと滞空時間を表示する.
- ボールの軌道をグラフ表示する.

ひな形

足りない部分もあるのでプログラムのコメントをよく読むこと
リンク切れの場合はその講義内でのアナウンスを参照すること。

NumMeth.h
https://drive.google.com/file/d/11iI6uiDu0BwQJMZ3n0uiIiR9DbXnqGEY/view

Matrix.h
https://drive.google.com/file/d/1M01tQ5-4I_Ip0oIPYy4RbZqXSjofYau0/view

```
// baseball.cpp: オイラー法を用いて野球ボールの軌道を計算するプログラム
#include "NumMeth.h"
using namespace std;
int main() {

   //* ボールの初期位置及び初期速度を設定する.
   double y1, speed, theta;
   double r1[2+1], v1[2+1], r[2+1], v[2+1], accel[2+1];
   cout << "高さの初期値(メートル) : "; cin >> y1;
   r1[1] = 0; r1[2] = y1;    // 初期位置ベクトル
   cout << "初期速度(m/s) : "; cin >> speed;
   cout << "初期角度(度) : "; cin >> theta;
   const double pi = 3.141592654;
   v1[1] = speed*cos(theta*pi/180);   // 初期速度 (x)
   v1[2] = speed*sin(theta*pi/180);   // 初期速度 (y)
   r[1] = r1[1]; r[2] = r1[2];   // 初期位置および初期速度を設定
   v[1] = v1[1]; v[2] = v1[2];

   //* 物理パラメータを設定(質量, Cd 値など)
   double Cd = 0.35;       // 空気抵抗(無次元)
   double area = 4.3e-3;   // 投射物の横断面積(m^2)
   double grav = 9.81;     // 重力加速度(m/s^2)
   double mass = 0.145;    // 投射物の質量(kg)
   double airFlag, rho;
   cout << "空気抵抗(あり:1, なし:0) : "; cin >> airFlag;
   if( airFlag == 0 )
      rho = 0;   // 空気抵抗なし
   else
      rho = 1.2;   // 空気の密度(kg/m^3)
   double air_const = -0.5*Cd*rho*area/mass;   // 空気抵抗定数

   //* ボールが地面に着くまで, あるいは最大の刻み数になるまでループ
   double tau;
   cout << "時間刻み τ(秒) : "; cin >> tau;
   int iStep, maxStep = 1000;   // 最大の刻み数
   double *xplot, *yplot, *xNoAir, *yNoAir;
   xplot = new double [maxStep + 1];
   yplot = new double [maxStep + 1];
   xNoAir = new double [maxStep + 1];
   yNoAir = new double [maxStep + 1];
   for( iStep=1; iStep<=maxStep; iStep++ ) {

      //* プロット用に位置(計算値および理論値)を記録する
      xplot[iStep] = r[1]; // プロット用に軌道を記録
      yplot[iStep] = r[2];
      double t = ( iStep-1 )*tau; // 現在時刻
      xNoAir[iStep] = r1[1] + v1[1]*t; // 位置(x)
      yNoAir[iStep] = 1; // 位置(y) ここに位置を求める式を書く
      //* ボールの加速度を計算する
```

```
        double normV = sqrt( v[1]*v[1] + v[2]*v[2] );
        accel[1] = air_const*normV*v[1];   // 空気抵抗
        accel[2] = air_const*normV*v[2];   // 空気抵抗
        accel[2] -= grav;                  // 重力

        //* オイラー法を用いて，新しい位置および速度を計算する

        /*
          ここにオイラー法のアルゴリズムを書く
        */


        //* ボールが地面に着いたら(y < 0)ループを抜ける

        /*
          if 文を使って書く
        */

    }

    //* 最大到達高さと滞空時間を表示する(このままだと正しい結果ではない)
    cout << "最大到達高さは" << r[2] << "メートル" << endl;
    //     滞空時間の表示をここに書く

    //* プロットする変数を出力する
    //   xplot, yplot xNoAir, yNoAir
    ofstream xplotOut("xplot.txt"), yplotOut("yplot.txt"),
        xNoAirOut("xNoAir.txt"), yNoAirOut("yNoAir.txt");

    int i;
    for( i=1; i<=iStep+1; i++ ) {
        xplotOut << xplot[i] << endl;
        yplotOut << yplot[i] << endl;
    }
    for( i=1; i<=iStep+1; i++ ) {
        xNoAirOut << xNoAir[i] << endl;
        yNoAirOut << yNoAir[i] << endl;
    }

    delete [] xplot, yplot, xNoAir, yNoAir;  // メモリを開放
}
```

↥

問題:3

時間刻みτの設定によって，何がどう変わるか説明せよ．

問題:4

ボールの軌道のグラフを gnuplot で出力せよ.
　なお，時間刻み τ は $0 < t \leq 2$ の間で 7 個選ぶこと.
　補助的に Octave を用いてもよい.

問題:5

オイラー法よりも高精度な数値計算アルゴリズムについて調べて、そのアルゴリズムを説明せよ．余力のある人は，アルゴリズムを実装しその結果をオイラー法と比較してみよ。

レポートに関して †

課題 †

実験テキスト中にある問題に回答せよ.

注意事項 †

- レポートは LaTeX で作成し、PDF ファイル(電子データ)と印刷物(紙媒体)を提出すること(手書き厳禁！)
- 必要なグラフや図は gnuplot や tgif 等のソフトを使うこと
- ページ番号・図番号・表番号は必ず記載すること
- 図や表は、該当する図や表の番号を引用して、文章中において説明すること
- プログラムがある場合には、ソースコードと、その説明(コメント)を入れること
- 「実験結果」と「考察」を必ず入れること
- レポートの表紙には以下の項目を記述すること
- 実験テーマ名、担当教員名
- 氏名、学籍番号
- 実験日、提出期限日、実際に提出した日 †

提出期限

実験終了後 1 週間以内(詳細は下記参照) ※厳守!!

ファイル名
slab4-1-[学籍番号].pdf
例:学籍番号 165701J の場合
　Slab4-1-165701J.pdf
　※アルファベットも忘れずに

メール件名

知能情報基礎演習4-1-slab-[学籍番号]
※アルファベットも忘れずに

レポート提出先

handout@nc.ie.u-ryukyu.ac.jp