

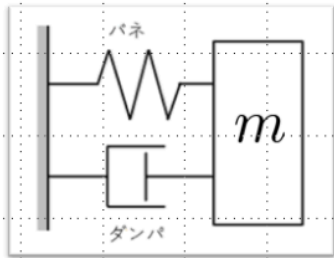
# 情報処理演習 II (課題4-1)

## BQ23107 窪田大輝

### 1. 目的

粘性抵抗中のバネ振動のシミュレーションをオイラー法と中点法で比較せよ。

個人課題4-1 (必須) 粘性抵抗中バネ振動のシミュレーションを  
オイラー法と中点法で比較せよ。



$$m \frac{d^2 x}{dt^2} = -kx - bv_x$$

提出締め切り：次週講義前

提出場所：[Scomb](#)/個人課題4-1 (粘性抵抗中振動)

提出物 1 (プログラムソース)：[SpringViscosityMid.c](#)

提出物 2 (レポート)：「学籍番号-氏名-課題4-1.拡張子」 ([Word](#), [PPT](#), [pdf](#)のどれか)

説明動画：<https://web.microsoftstream.com/video/c10f9d6f-a01a-4672-a677-a5fca9a1510a>

### 2. 結果(作成したプログラム)

#### 2.1. 講義動画で作成したプログラム

図1は講義動画で作成したSpringViscosityMid.cファイルである。このプログラムは、粘性抵抗中のバネ振動のシミュレーションを中点法で行っている。

```

#include <stdio.h>

const double m=3.0; // 質量 kg
const double k=50; // ばね定数 N/m
const double b=1; // 粘性抵抗係数 b = kg/s
//const double g=9.8; // 重力加速度 9.8 m/s^2

double f(double x, double vx){
    return -k*x - b*vx;
}

int main(){

    double t,x,vx,t0,t1,x0,vx0,dt,xx,ddt,k1,k2;
    int i,numi,nump;

    t0=0; //初期時刻
    t1=5; //終了時刻

    x0=3; //初期位置
    vx0=0; //初期速度

    dt=0.1;
    numi=(t1-t0)/dt;

    t=t0;
    x=x0;
    vx=vx0;

    FILE *gp;
    gp=popen("gnuplot --persist -slow","w"); //gnuplotの呼び出し
    FILE *fp;
    fp=fopen("SpringViscosityMid.dat","w"); //datファイルの作成

    for(i=1; i<=numi; i++){
        printf("%f %f\n",t,x);
        fprintf(fp, "%f %f\n",t,x); //datファイルに書き込み
    }
}

```

```

k1=x+(dt/2)*vx;    // 現在のxにdt/2だけ進んだときの位置
k2=vx+(dt/2)*f(x,vx)/m;    // 現在のvxにdt/2だけ進んだときの速度
                        // 与えられた位置と速度に対する加速度を計算する関数fを用いる
x=x+dt*k2;          //新しい位置xを計算する
                        //傾きk2にステップサイズdtをかけてxに足す
vx=vx+dt*f(k1,k2)/m;    //新しい速度vxを計算する
                        //位置k1と速度k2から加速度を計算
                        //ステップサイズdtをかけてvxに足す

t=t+dt;
}

fclose(fp);
fprintf(gp, "plot \"SpringViscosityMid.dat\" with lines\n"); //gnuplotでdatファイルをプロット
fflush(gp);
pclose(gp);

}

```

Fig.1 講義動画で作成したSpringViscosityMid.cのプログラム

## 2.2. オイラー法と中点法を比較したプログラム

オイラー法と中点法を比較するためのプログラムを作成した。その上でGnuplotでグラフを描画した。今回はプログラム上でdatファイルを生成するようにし、Gnuplotでグラフ描画も行った。

```

/*
課題4-1
オイラー法と中点法の比較
dt=0.1
*/
#include <stdio.h>

const double m=3.0; // 質量 kg
const double k=50; // ばね定数 N/m
const double b=1; // 粘性抵抗係数 b = kg/s
//const double g=9.8; // 重力加速度 9.8 m/s^2

double f(double x, double vx){
    return -k*x - b*vx;
}

int main(){

    double t,x,vx,t0,t1,x0,vx0,dt,xx,ddt,k1,k2;
    int i,numi,nump;

    t0=0;
    t1=5;

    x0=3;
    vx0=0;

    dt=0.0001;
    numi=(t1-t0)/dt;

    t=t0;
    x=x0;
    vx=vx0;

    FILE *gp;
    gp = popen("gnuplot --persist -slow","w");
    FILE *fp1;
    fp1 = fopen("SpringViscosity_euler.dat","w");
    FILE *fp2;

```

```

fp2 = fopen("SpringViscosity_midpoint.dat","w");

for(i=1; i<=numi; i++){
    printf("%f %f\n",t,x);
    fprintf(fp1, "%f %f\n",t,x);

    k1=x+(dt/2)*vx;
    k2=vx+(dt/2)*f(x,vx)/m;
    x=x+dt*k2;
    vx=vx+dt*f(k1,k2)/m;

    t=t+dt;
}

t=t0;
x=x0;
vx=vx0;

for(i=1; i<=numi; i++){
    printf("%f %f\n",t,x);
    fprintf(fp2, "%f %f\n",t,x);
    xx=x;
    x=x+dt*vx;
    vx=vx+dt*f(xx,vx)/m;
    t=t+dt;
}

fclose(fp1);
fclose(fp2);
fprintf(gp, "plot \"SpringViscosity_euler.dat\" with lines, \
        \"SpringViscosity_midpoint.dat\" with lines\n");
fprintf(gp, "set xlabel \"t\"\n");
fprintf(gp, "set ylabel \"x\"\n");
fprintf(gp, "set title \"Euler method and Midpoint method dt=%.4f\"\n", dt);
fprintf(gp, "replot\n");
fflush(gp);

```

```
pclose(gp);  
  
}
```

Fig.2 オイラー法と中点法を比較したプログラム

図2はオイラー法と中点法を比較したプログラムである。このプログラムは、粘性抵抗中のバネ振動のシミュレーションをオイラー法と中点法で行っている。

## 2.3. プログラムの実行結果

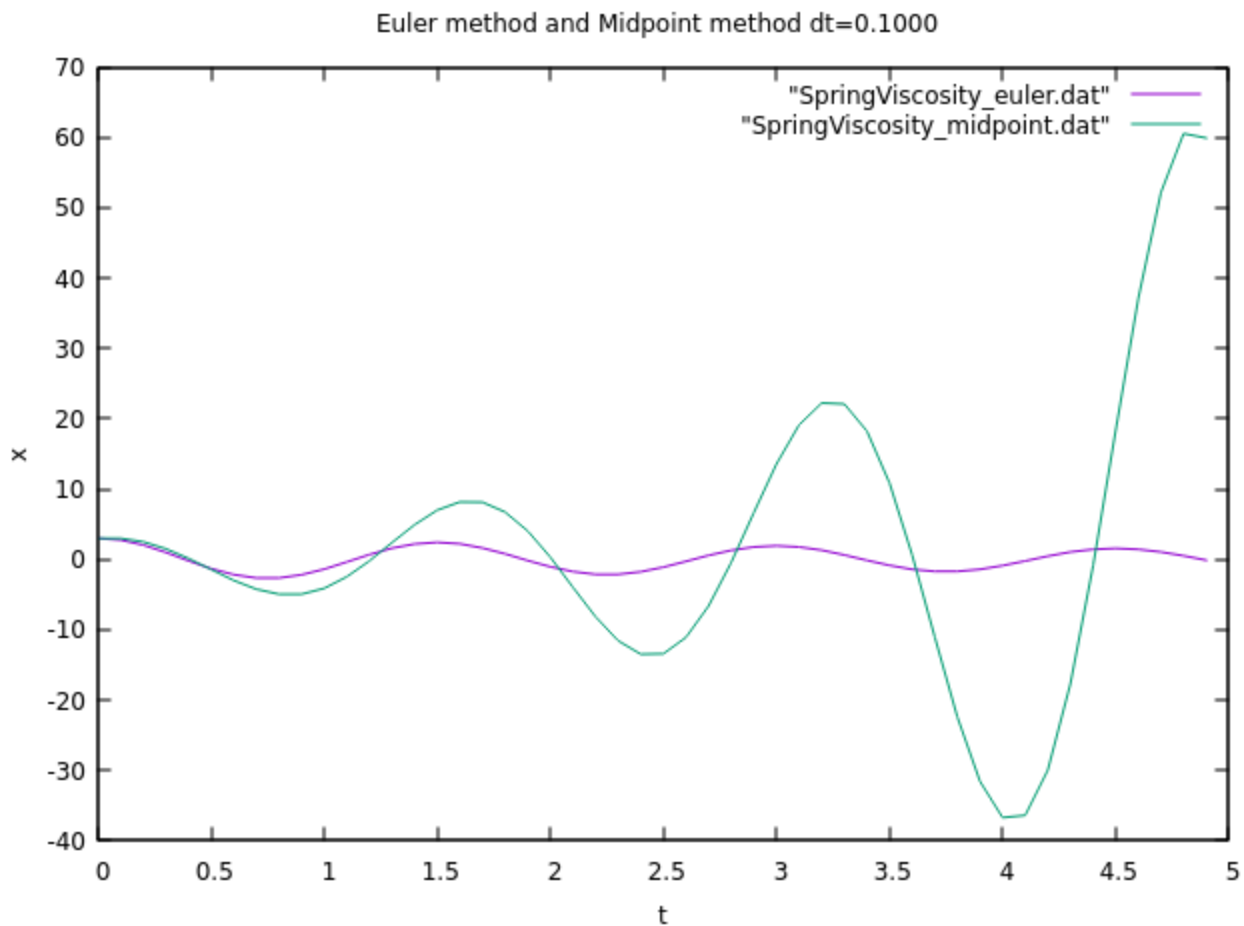


Fig.3  $dt=0.1$ の時のオイラー法と中点法の比較

図3は $dt = 0.1$ の時のオイラー法と中点法の比較である。粘性抵抗中のバネ振動であるため、減衰しなければならない。しかし、オイラー法では減衰すらせずに増幅している。一方、中点法では減衰している。

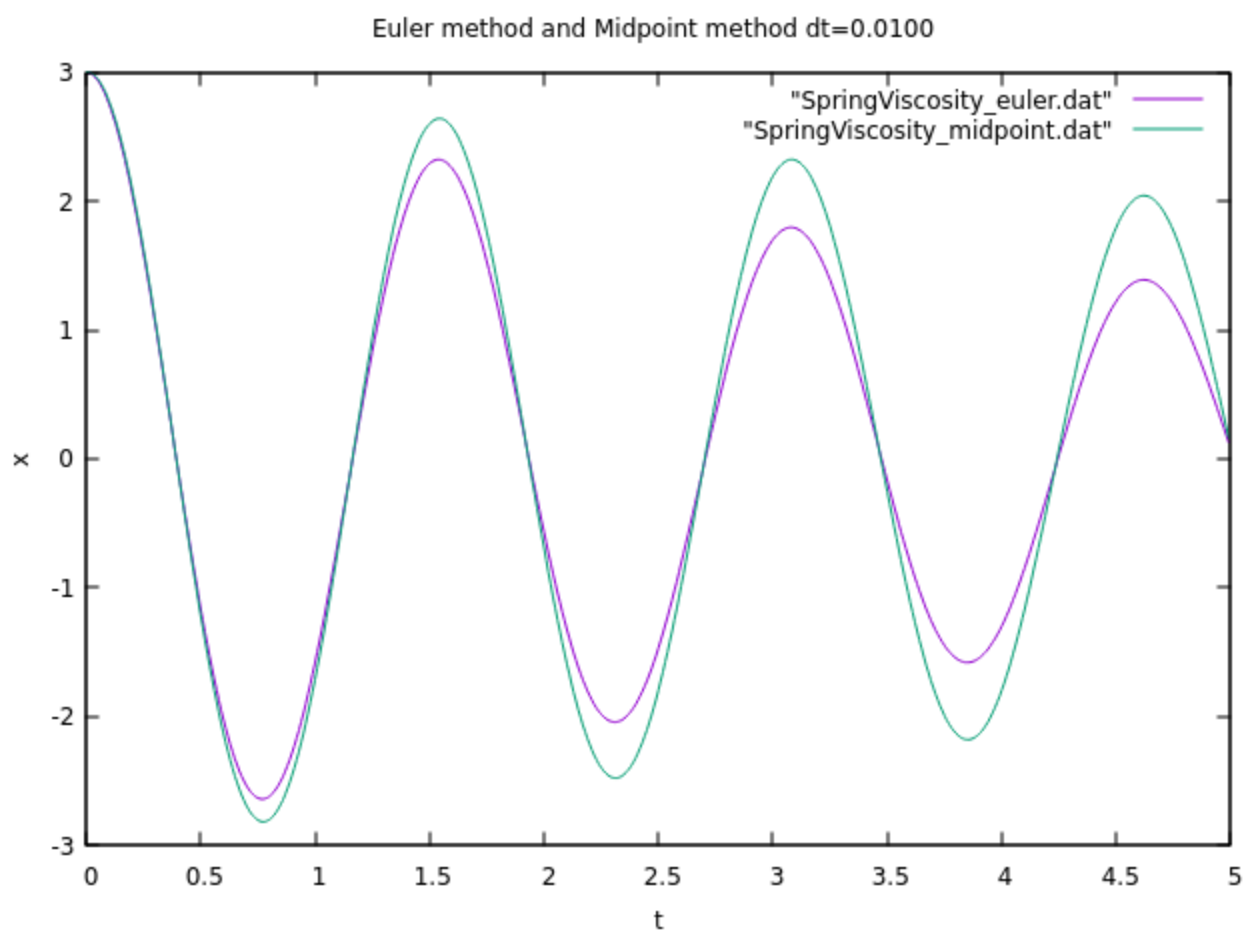


Fig.4  $dt=0.01$ の時のオイラー法と中点法の比較

図4は $dt = 0.01$ の時のオイラー法と中点法の比較である． $dt = 0.1$ の時とは異なり，オイラー法でも，中点法でも減衰している．しかし，オイラー法では中点法と比べて，明らかに誤差が蓄積している．

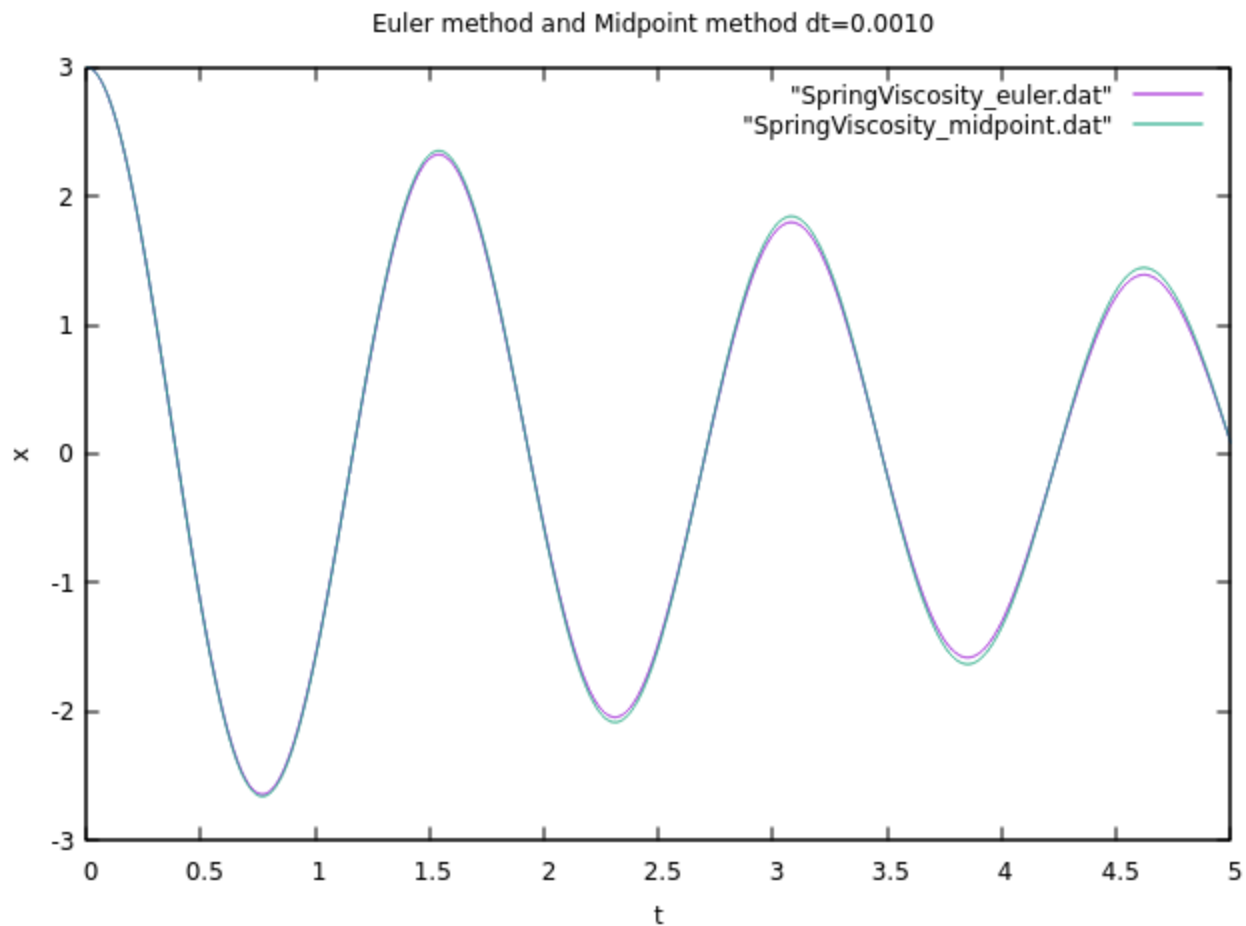


Fig.5  $dt=0.001$ の時のオイラー法と中点法の比較

時刻 $t$	オイラー法	中点法	差(中点法-オイラー法)
4.621000	1.388795	1.443330	0.054535
4.999000	0.091040	0.090518	-0.000522

Table 1  $dt=0.001$ の時の時刻 $t$ のオイラー法と中点法の比較

図5は $dt = 0.001$ の時のオイラー法と中点法の比較である． $dt = 0.01$ の時と比べて，オイラー法と中点法の差が小さくなっている．定性的に評価するためにdatファイルを参照する．表1のように $t = 4.621000$ の時にはオイラー法と中点法の差が0.054535である．また， $t = 4.999000$ の時にはオイラー法と中点法の差が $-0.000522$ である．



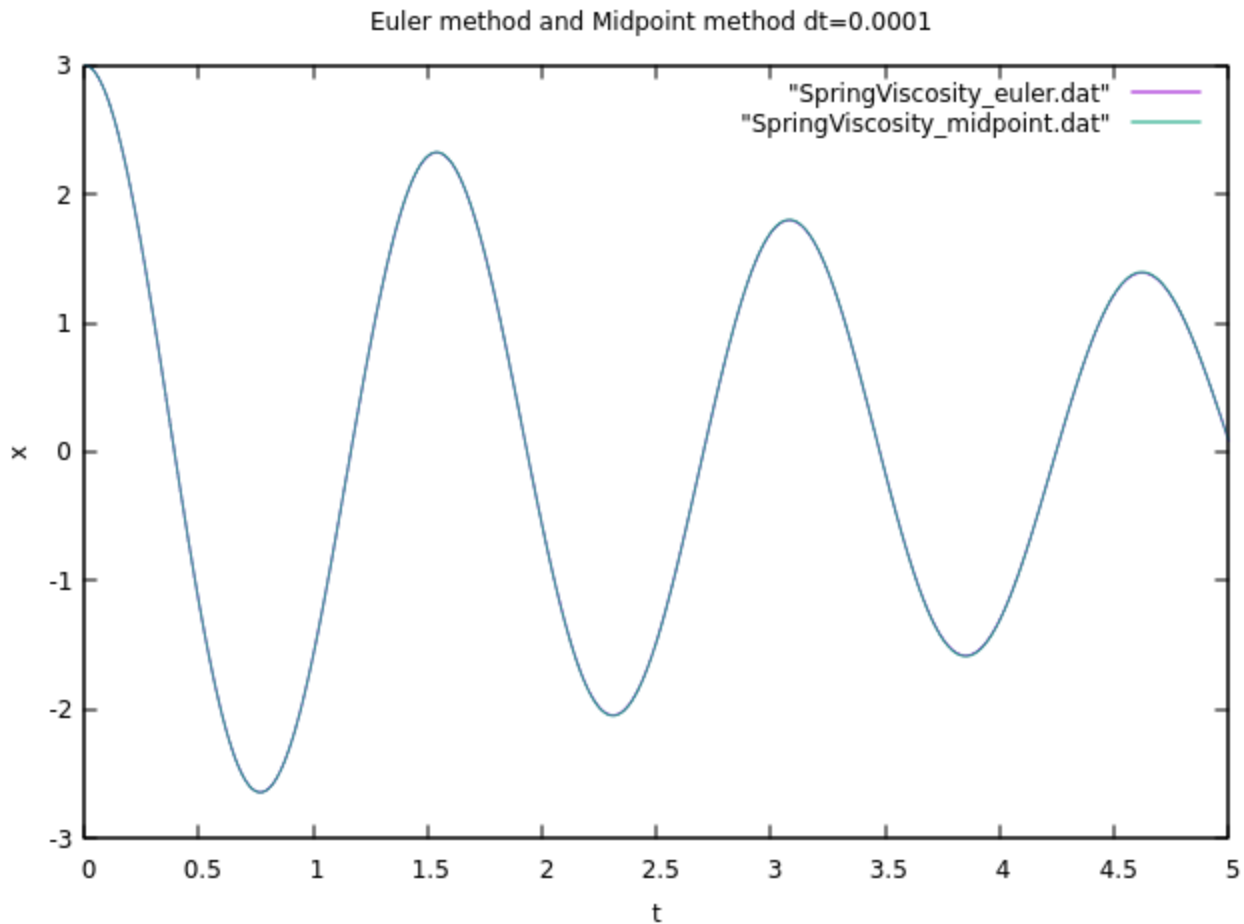


Fig.6  $dt=0.0001$ の時のオイラー法と中点法の比較

時刻 $t$	オイラー法	中点法	差(中点法-オイラー法)
4.621000	1.388800	1.394159	0.005359
4.999900	0.086320	0.086237	-0.000083

Table 2  $dt=0.0001$ の時の時刻 $t$ のオイラー法と中点法の比較

図6は $dt = 0.0001$ の時のオイラー法と中点法の比較である． $dt = 0.001$ の時と比べて，オイラー法と中点法の差が小さくなっている．定性的に評価するためにdatファイルを参照する．表2のように $t = 4.621000$ の時にはオイラー法と中点法の差が0.005359である．また， $t = 4.999900$ の時にはオイラー法と中点法の差が $-0.000083$ である．

### 3. 考察

Gnuplotで描画した結果と生成されたdatファイルを踏まえて考察を行う。

$dt = 0.01$ である図3では明らかに近似として使えない。

時刻 $t$	刻み幅 $dt$	オイラー法	中点法	差(中点法-オイラー法)
4.621000	0.001	1.388795	1.443330	0.054535
4.621000	0.0001	1.388800	1.394159	0.005359
4.999000	0.001	0.091040	0.090518	-0.000522
4.999900	0.0001	0.086320	0.086237	-0.000083

Table 3  $dt=0.0001$ の時の時刻 $t$ のオイラー法と中点法の比較

表3はオイラー法と中点法の比較をまとめた表である。 $dt = 0.001$ の時と比べて、 $dt = 0.0001$ の時の方がオイラー法と中点法の差が小さくなっている。 $dt = 0.0001$ は $t = 5$ の時点で最大でも誤差が $1/100$ 以下になっている。

しかし、オイラー法であっても中点法であっても誤差は蓄積するので、自分の必要な範囲に誤差が収まるように刻み幅を決める必要がある。無駄に細かくても実行に時間もメモリも使うため、適切な刻み幅を決める必要がある。特に時刻 $t$ が増加すると誤差が大きくなるため、 $t$ が大きくなるような場合は刻み幅を小さくしなければならない。