情報処理演習グループ課題2

役職	学籍番号	名前	備考
リーダー	BQ23107	窪田 大輝	
プログラムリーダー	BQ23008	脇家 優太	
企画係	BQ23071	平川 奨	
連絡係	BQ23060	塚田 水月	
書記１	BQ23110	信賀 晃	
書記２	BQ23103	山田 泰我	欠席

情報処理演習グループ課題2
仮説(運動学として解く)←信賀
高さhを増加させる場合
▪高さ0の時
▪高さがhの時

▪粘性抵抗を考慮して高さhを増加させる場合
結果(C言語で球値)
高さhを増加させる場合
▪最大飛距離を計算
▪最大飛距離をgnuplotでプロット
粘性抵抗を考慮して高さhを増加させる場合
▪最大飛距離を計算
▪最大飛距離をgnuplotでプロット

1. 仮説(運動学として解く)←信賀

1.1. 高さhを増加させる場合

1.1.1.高さ0の時

z座標0.0，初速$v_0 = 10$，角度θで斜方投射した時に最大飛距離になるθを求める．
重力加速度を$g = 9.8$とする．

水平方向の初期速度：$v_x(t) = v0\cos(\theta)$
鉛直方向の初期速度：$v_z(t) = v0\sin(\theta)$

水平方向の初期位置：$x(t) = 0$
鉛直方向の初期位置：$z(t) = 0$

水平方向の運動方程式：$a_x(t) = 0$
鉛直方向の運動方程式：$a_z(t) = -g$

水平方向の速度：$v_x(t) = v0\cos(\theta)$
鉛直方向の速度：$v_z(t) = v0\sin(\theta) - gt$

水平方向の位置：$x(t) = v0\cos(\theta)t$
鉛直方向の位置：$z(t) = v0\sin(\theta)t - \frac{1}{2}gt^2$

高さが0になる時間tを求める．

$$
\begin{aligned}
z(t) &= 0 \\
v_0\sin(\theta)t - \frac{1}{2}gt^2 &= 0 \\
t &= \frac{2v_0\sin(\theta)}{g}
\end{aligned}
$$

このtを$x(t)$に代入すると，最大飛距離xが求まる．

$$
\begin{aligned}
x(t) &= v_0 \cos(\theta) t \\
&= v_0 \cos(\theta) \frac{2v_0 \sin(\theta)}{g} \\
&= \frac{2v_0^2 \sin(\theta)\cos(\theta)}{g} \\
&= \frac{v_0^2 \sin(2\theta)}{g}
\end{aligned}
$$

$0^\circ < \theta < 90^\circ$の範囲で最大値を求める.
$sin(90^\circ) = 1$で最大値なので，$\theta = 45^\circ$の時に最大飛距離になる.

1.1.2.高さがhの時

水平方向の初期速度：$v_x(0) = v0\cos(\theta)$
鉛直方向の初期速度：$v_z(0) = v0\sin(\theta)$

水平方向の初期位置：$x(0) = 0$
鉛直方向の初期位置：$z(0) = h$

水平方向の運動方程式：$a_x(t) = 0$
鉛直方向の運動方程式：$ma_z(t) = -mg$

水平方向の速度：$v_x(t) = v0\cos(\theta)$
鉛直方向の速度：$v_z(t) = v0\sin(\theta) - gt$

水平方向の位置：$x(t) = v0\cos(\theta)t$
鉛直方向の位置：$z(t) = v0\sin(\theta)t - \frac{1}{2}gt^2 + h$

高さが0になる時間tを求める.

$$
\begin{aligned}
z(t) &= 0 \\
v_0 \sin(\theta) t - \frac{1}{2} g t^2 + h &= 0 \\
t &= \frac{v_0 \sin(\theta) \pm \sqrt{v_0^2 \sin^2(\theta) - gh}}{g}
\end{aligned}
$$

このtを$x(t)$に代入すると，最大飛距離xが求まる．

$$
\begin{aligned}
x(t) &= v_0 \cos(\theta) t \\
&= v_0 \cos(\theta) \frac{v_0 \sin(\theta) \pm \sqrt{v_0^2 \sin^2(\theta) - gh}}{g} \\
&= \frac{v_0^2 \sin(\theta)\cos(\theta) \pm v_0 \cos(\theta) \sqrt{v_0^2 \sin^2(\theta) - gh}}{g} \\
&= \frac{v_0^2 \sin(2\theta) \pm v_0 \cos(\theta) \sqrt{v_0^2 \sin^2(\theta) - gh}}{g}
\end{aligned}
$$

この式にhを実数を代入する．その後$x(t)$が最大となるようにθを求めれば良い．

1.2. 粘性抵抗を考慮して高さhを増加させる場合

水平方向の初期速度：$v_x(0) = v_0 \cos(\theta)$
鉛直方向の初期速度：$v_z(0) = v_0 \sin(\theta)$

水平方向の初期位置：$x(0) = 0$
鉛直方向の初期位置：$z(0) = h$

粘性抵抗係数$b = 2.5$

水平方向の運動方程式：$ma_x(t) = m\frac{dv_x}{dt} = -bv_x(t)$
鉛直方向の運動方程式：$ma_z(t) = m\frac{dv_z}{dt} = -mg - bv_z(t)$

水平方向の速度：

$$
\begin{aligned}
\frac{dv_x}{dt} &= -\frac{b}{m} v_x(t) \\
\int \frac{1}{v_x(t)}\, dv_x &= -\int \frac{b}{m}\, dt \\
\ln |v_x(t)| &= -\frac{b}{m} t + C \\
v_x(t) &= e^{-\frac{b}{m}t + C} \\
v_x(t) &= Ae^{-\frac{b}{m}t}
\end{aligned}
$$

初速度が$v_x(0)$なので$A = v_x(0) = v_0 \cos(\theta)$より

$$v_x(t) = v_0 \cos(\theta) e^{-\frac{b}{m}t}$$

鉛直方向の速度：

$$\begin{aligned}
\frac{dv_z}{dt} &= -g - \frac{b}{m} v_z(t) \\
\int \frac{1}{g + \frac{b}{m} v_z(t)} \, dv_z &= -\int dt \\
\int \left(1 - \frac{\frac{b}{m}}{g} v_z(t) \right) \, dv_z &= -\int dt \\
v_z(t) - \frac{\frac{b}{m}}{g} v_z^2(t) &= -t + C
\end{aligned}$$

初速度が$v_z(0)$なので$C = v_z(0) = v_0 \sin(\theta)$より...
と解きたいが，解けないので数値計算や近似法を用いることが一般的である.

2. 結果(C言語で球値)

2.1. 高さhを増加させる場合

2.1.1.最大飛距離を計算

図1は，高さをz軸座標を0から刻み幅1.0ごとに10.0まで増加させた時に，θを変化させて，何度で投げると飛距離を最大にできるかを計算するプログラムである.
図2はその結果である.

```c
#include <stdio.h>
#include <math.h>
#define Gravity -9.8

#ifndef M_PI
#define M_PI 3.141592
#endif

int main(void) {;

    double x,z,vx0,vz0,vx,vz,theta;
    double z_min = 0.0;   // z座標初期位置
    double z_max = 10.0; // z座標最終位置
    double delta_z = 1.0; // z座標刻み幅
    double z0; //z座標の仮置き変数

    double theta_min = 0.0; // theta初期角度
    double theta_max = 50.0; // theta最終座標
    double delta_theta = 0.01; // theta刻み幅

    double delta_t = 0.01; // t刻み幅

    double v0 = 10.0; // 初期速度


    FILE *fp;
    fp = fopen("d1.dat", "w");

    for (z0 = z_min; z0 < z_max; z0 += delta_z) {
        double theta_flag = 0.0;
        double x_max = 0; // 初期
        for (theta = theta_min; theta < theta_max; theta += delta_theta) {
            x = 0;
            z = z0;
            vx0 = v0 * cos(theta / 180.0 * M_PI);
            vz0 = v0 * sin(theta / 180.0 * M_PI);
            vx = vx0;
            vz = vz0;
            int count = 0;
```

```
            //printf("%f %f %f\n",theta ,vx, vz);
            while(z>=0.0){
                vx += 0 * delta_t;
                x += vx * delta_t;
                vz += Gravity * delta_t;
                z += vz * delta_t;
                count += 1;
            }

            if (x_max < x) {
                x_max = x;
                theta_flag = theta;
            }

        }

        printf("z=%.3fの時，theta=%.3fで最大飛距離x=%.3fである．\n", z0, theta_flag, x_max);
        fprintf(fp, "%f %f\n", z0, theta_flag);
    }

    return 0;
}
```

Fig.1 Z座標と角度をパラメータとした時の，最大飛距離を求めるC言語プログラム

```
0.000000 44.880000
1.000000 42.320000
2.000000 40.310000
3.000000 38.290000
4.000000 36.610000
5.000000 35.220000
6.000000 34.080000
7.000000 33.160000
8.000000 32.420000
9.000000 30.890000
```

Fig.2 Z座標と角度をパラメータとした時の，最大飛距離を求めるC言語プログラムの実行結果

2.1.2.最大飛距離をgnuplotでプロット

図3は図1のプログラムで球値した最大飛距離になるまでの軌跡をdatファイルに格納するプログラムである．datファイルは複数に出力されるので，その結果を統合してGnupplotで描画したプログラムが図4である．

```c
#include <stdio.h>
#include <math.h>
#define Gravity -9.8

#ifndef M_PI
#define M_PI 3.141592
#endif

int main(void) {
    double x, z, vx0, vz0, vx, vz, theta;
    double z_min = 0.0;
    double z_max = 10.0; // 30.0
    double delta_z = 1.0;
    double z0;

    double theta_min = 0.0;
    double theta_max = 50.0;
    double delta_theta = 0.1;

    double delta_t = 0.01; //0.01

    double v0 = 10.0;
    double b = 2.5;
    double m = 3.0;

    FILE *fp;
    fp = fopen("d2.dat", "r");
    if (fp == NULL) {
        printf("ファイルを開けませんでした. \n");
        return 1;
    }

    double z0_value, theta_value;
    int count = 1;

    while (fscanf(fp, "%lf %lf", &z0_value, &theta_value) != EOF) {
        printf("z=%fの時, theta=%f\n", z0_value, theta_value);
        double theta_flag = 0.0;
        double x_max = 0;
```

```
        x = 0;
        z = z0_value;
        vx0 = v0 * cos(theta_value / 180.0 * M_PI);
        vz0 = v0 * sin(theta_value / 180.0 * M_PI);
        vx = vx0;
        vz = vz0;

        char filename[20];
        sprintf(filename, "data2_%d.dat", count);
        FILE *fp2 = fopen(filename, "w");

        if (fp2 == NULL) {
            printf("%sファイルを開けませんでした. \n", filename);
            return 1;
        }

        int inner_count = 0;

        while (z >= 0.0) {
            printf("     %d %.3f %.3f\n", inner_count, x, z);
            fprintf(fp2, "%f %f\n", x, z);
            vx = vx + delta_t * (0 - b * vx / m);
            x += vx * delta_t;
            vz = vz + delta_t * (Gravity - b * vz / m);
            z += vz * delta_t;
            inner_count += 1;
        }

        fclose(fp2); // ファイルを閉じる
        count += 1;  // カウントを更新
    }

    fclose(fp); // ファイルを閉じる

    return 0;
}
```

Fig.3 最大飛距離になる時の軌跡をdatファイルに格納するプログラム

```c
#include <stdio.h>
//画像を保存

int main() {
    FILE *gnuplotPipe = popen ("gnuplot -persistent", "w");

    if (gnuplotPipe == NULL) {
        printf("Error opening pipe to GNU plot.\n");
        return 1;
    }

    fprintf(gnuplotPipe, "set title 'projectile plot'\n");
    fprintf(gnuplotPipe, "set xrange [0:18]\n");
    fprintf(gnuplotPipe, "set yrange [0:18]\n");


    for (int i = 1; i <= 10; i++) {
        if (i == 1) {
            fprintf(gnuplotPipe, "plot 'data1_%d.dat' with lines\n", i);
        } else {
            fprintf(gnuplotPipe, "replot 'data1_%d.dat' with lines\n", i);
        }
    }
    fprintf(gnuplotPipe, "replot\n");


    fflush(gnuplotPipe);
    fclose(gnuplotPipe);
    return 0;
}
```

Fig.4 複数datファイルから軌跡を描くプログラム

図5は図4で示したプログラムにより生成されたグラフである.

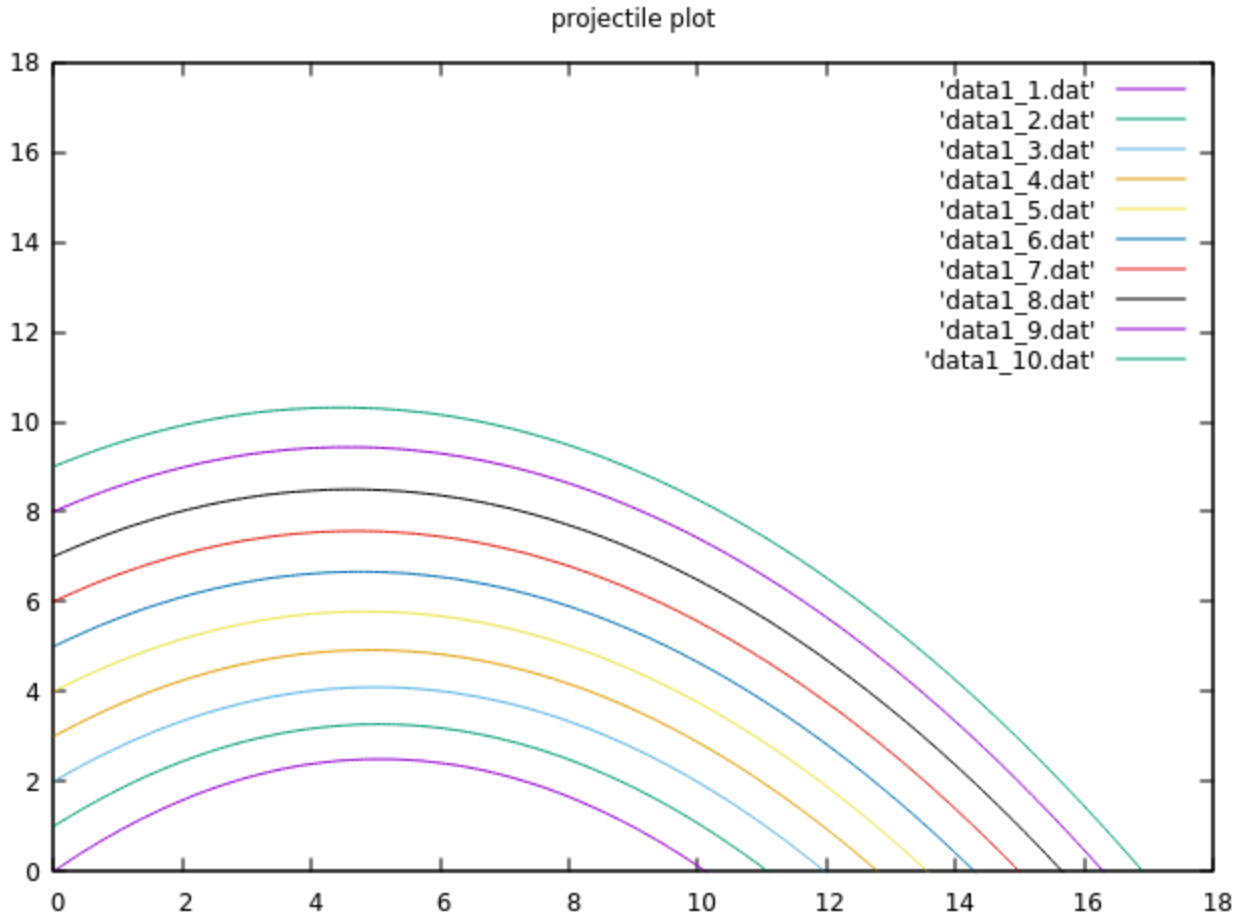

Fig.5 図4によって生成されたグラフ

2.2. 粘性抵抗を考慮して高さhを増加させる場合

2.2.1.最大飛距離を計算

図6は，高さをz軸座標を0から刻み幅1.0ごとに10.0まで増加させた時に，θを変化させて，何度で投げると飛距離を最大にできるかを計算するプログラムである．
図7はその結果である．

```c
#include <stdio.h>
#include <math.h>
#define Gravity -9.8

#ifndef M_PI
#define M_PI 3.141592
#endif

int main(void) {;

    double x,z,vx0,vz0,vx,vz,theta;
    double z_min = 0.0;   // z座標初期位置 0.0
    double z_max = 10.0; // z座標最終位置 30.0
    double delta_z = 1.0; // z座標刻み幅
    double z0; //z座標の仮置き変数

    double theta_min = 0.0; // theta初期角度
    double theta_max = 50.0; // theta最終座標
    double delta_theta = 0.1; // theta刻み幅

    double delta_t = 0.01; // t刻み幅 0.01

    double v0 = 10.0; // 初期速度
    double b = 2.5; // 粘性抵抗係数
    double m = 3.0; // 質量



    FILE *fp;
    fp = fopen("d2.dat", "w");

    for (z0 = z_min; z0 < z_max; z0 += delta_z) {
        double theta_flag = 0.0;
        double x_max = 0; // 初期
        for (theta = theta_min; theta < theta_max; theta += delta_theta) {
            x = 0;
            z = z0;
            vx0 = v0 * cos(theta / 180.0 * M_PI);
            vz0 = v0 * sin(theta / 180.0 * M_PI);
            vx = vx0;
```

```
            vz = vz0;
            int count = 0;
            //printf("%f %f %f\n",theta ,vx, vz);
            while(z>=0.0){
                vx = vx + delta_t * (0 - b * vx/m);
                x += vx * delta_t;
                vz = vz + delta_t * (Gravity - b * vz/m);
                //vz+dt*(g - b*vz/m)
                z += vz * delta_t;
                count += 1;
                //printf("%f %d %.3f %.3f\n", theta,count, x, z);
            }

            if (x_max < x) {
                x_max = x;
                theta_flag = theta;
            }

        }

        printf("z=%.3fの時，theta=%.3fで最大飛距離x=%.3fである．\n", z0, theta_flag, x_max);
        fprintf(fp, "%f %f\n", z0, theta_flag);
    }

    return 0;
}
```

```
0.000000 36.800000
1.000000 32.300000
2.000000 27.500000
3.000000 24.500000
4.000000 22.100000
5.000000 18.600000
6.000000 16.500000
7.000000 15.800000
8.000000 13.700000
9.000000 12.900000
```

Fig.7 Z座標と角度をパラメータとした時の，最大飛距離を求めるC言語プログラムの実行結果

2.2.2.最大飛距離をgnuplotでプロット

図8は図6のプログラムで球値した最大飛距離になるまでの軌跡をdatファイルに格納するプログラムである．datファイルは複数に出力されるので，その結果を統合してGnupplotで描画したプログラムが図4である．

```c
#include <stdio.h>
#include <math.h>
#define Gravity -9.8

#ifndef M_PI
#define M_PI 3.141592
#endif

int main(void) {
    double x, z, vx0, vz0, vx, vz, theta;
    double z_min = 0.0;
    double z_max = 10.0; // 30.0
    double delta_z = 1.0;
    double z0;

    double theta_min = 0.0;
    double theta_max = 50.0;
    double delta_theta = 0.1;

    double delta_t = 0.01; //0.01

    double v0 = 10.0;
    double b = 2.5;
    double m = 3.0;

    FILE *fp;
    fp = fopen("d2.dat", "r");
    if (fp == NULL) {
        printf("ファイルを開けませんでした. \n");
        return 1;
    }

    double z0_value, theta_value;
    int count = 1;

    while (fscanf(fp, "%lf %lf", &z0_value, &theta_value) != EOF) {
        printf("z=%fの時, theta=%f\n", z0_value, theta_value);
        double theta_flag = 0.0;
        double x_max = 0;
```

```
        x = 0;
        z = z0_value;
        vx0 = v0 * cos(theta_value / 180.0 * M_PI);
        vz0 = v0 * sin(theta_value / 180.0 * M_PI);
        vx = vx0;
        vz = vz0;

        char filename[20];
        sprintf(filename, "data2_%d.dat", count);
        FILE *fp2 = fopen(filename, "w");

        if (fp2 == NULL) {
            printf("%sファイルを開けませんでした. \n", filename);
            return 1;
        }

        int inner_count = 0;

        while (z >= 0.0) {
            printf("     %d %.3f %.3f\n", inner_count, x, z);
            fprintf(fp2, "%f %f\n", x, z);
            vx = vx + delta_t * (0 - b * vx / m);
            x += vx * delta_t;
            vz = vz + delta_t * (Gravity - b * vz / m);
            z += vz * delta_t;
            inner_count += 1;
        }

        fclose(fp2); // ファイルを閉じる
        count += 1;  // カウントを更新
    }

    fclose(fp); // ファイルを閉じる

    return 0;
}
```

Fig.8 最大飛距離になる時の軌跡をdatファイルに格納するプログラム

```c
#include <stdio.h>
//画像を保存

int main() {
    FILE *gnuplotPipe = popen ("gnuplot -persistent", "w");

    if (gnuplotPipe == NULL) {
        printf("Error opening pipe to GNU plot.\n");
        return 1;
    }

    fprintf(gnuplotPipe, "set title 'projectile resistance plot'\n");
    fprintf(gnuplotPipe, "set xrange [0:10]\n");
    fprintf(gnuplotPipe, "set yrange [0:10]\n");


    for (int i = 1; i <= 10; i++) {
        if (i == 1) {
            fprintf(gnuplotPipe, "plot 'data2_%d.dat' with lines\n", i);
        } else {
            fprintf(gnuplotPipe, "replot 'data2_%d.dat' with lines\n", i);
        }
    }
    fprintf(gnuplotPipe, "replot\n");


    fflush(gnuplotPipe);
    fclose(gnuplotPipe);
    return 0;
}
```

Fig.9 複数datファイルから軌跡を描くプログラム

図10は図9で示したプログラムにより生成されたグラフである.

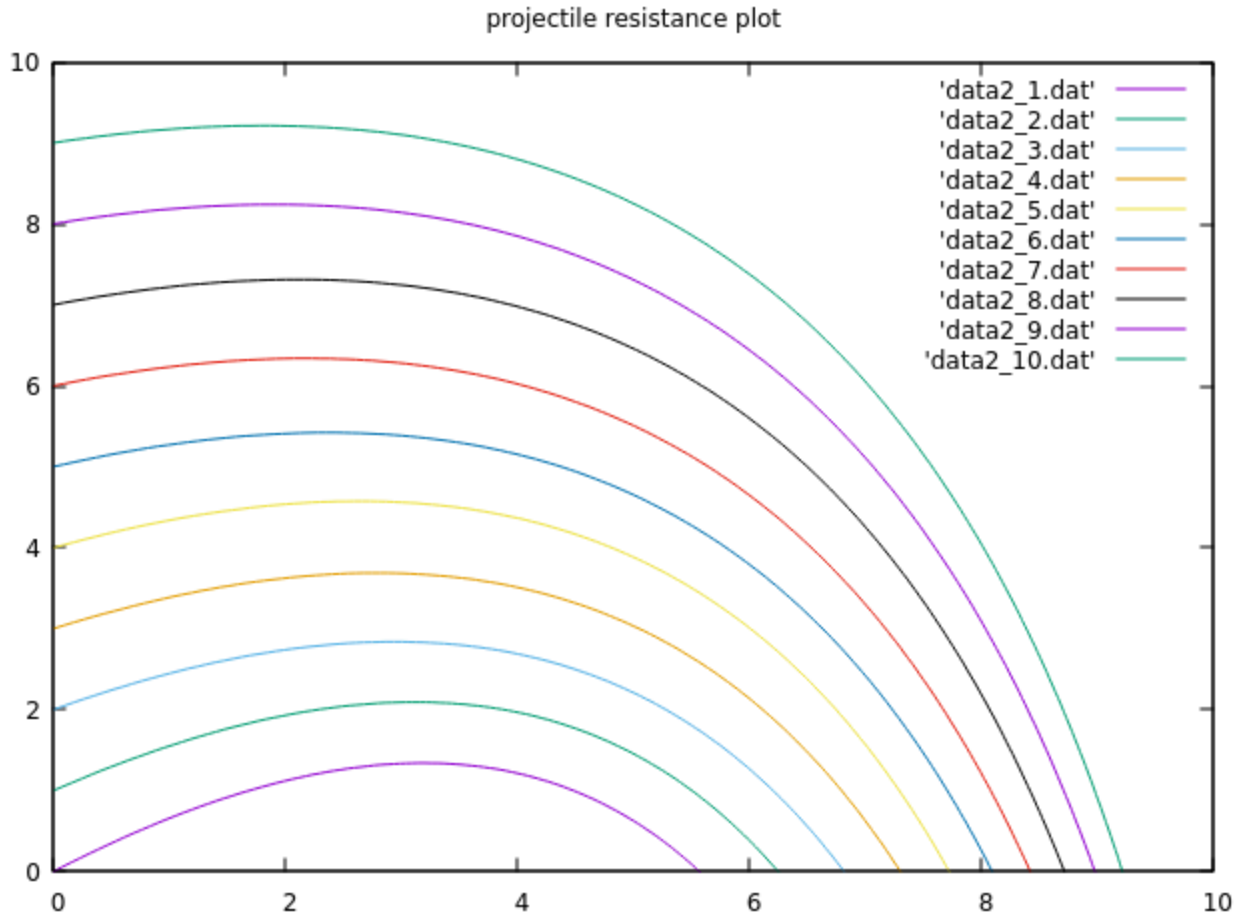

Fig.10 図4によって生成されたグラフ

仮説で論じた通り，計算が難しいものも，オイラー法により計算できた.