

アプリケーションのインストール

まず、圧縮フォルダ mingw.zip をすべて展開して、中身を確認する。

①フォルダ MinGW を C:¥にコピーする。

②フォルダ emacs-24.3 を C:¥にコピーする。

③フォルダ gnuplot を C:¥にコピーする。

④システム環境変数を設定する：

スタートボタン→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→詳細設定→環境変数

「システム環境変数」の中の「Path」を選択する。

「編集」をクリックする。

「;」で区切りながら、「C:¥MinGW¥bin;C:¥emacs-24.3¥bin;C:¥gnuplot¥bin」を追加する。

⑤C:¥MinGW¥msys¥1.0¥msys.bat をダブルクリックして実行する。

ターミナルが出てきたら以下のコマンドを実行する。

```
$ exit
```

⑥ショートカットを作成する：

C:¥MinGW¥msys¥1.0¥msys.bat へのショートカットと

C:¥MinGW¥msys¥1.0¥home¥username へのショートカットを
デスクトップ上に作成する。

⑦フォルダ files の中の profile と emacs を

C:¥MinGW¥msys¥1.0¥home¥username¥の中にコピーする。

⑧ショートカットから msys を開始する。

ターミナルが出てきたら以下のコマンドを実行する。

```
$ mv profile .profile
```

```
$ mv emacs .emacs
```

```
$ exit
```

簡単操作マニュアル

ターミナルの使い方

ターミナルを起動する	デスクトップの「msys. bat へのショートカット」をダブルクリック
①ファイルの一覧を表示する	ls
ファイルをコピーする	cp <u>コピー元ファイル名</u> <u>コピー先ファイル名</u>
ファイル名を変更する	mv <u>変更前のファイル名</u> <u>変更後のファイル名</u>
ファイルを消去する	rm <u>ファイル名</u>
②E m a c s を起動する	emacs &
③コンパイルする	cc <u>ファイル名</u> 必要に応じて: cc <u>ファイル名</u> -lm
④実行する	./a.out
⑤ファイルの中身を見る	less <u>ファイル名</u> 矢印キーで移動、Qを押して終了
⑥g n u p l o t を起動する	gnuplot
ターミナルを終了する	exit

E m a c s の使い方

①ファイルを開く	コントロールキー+X コントロールキー+F →ファイル名を入力
②ファイルを保存する	コントロールキー+X コントロールキー+S
E m a c s を終了する	コントロールキー+X コントロールキー+C

g n u p l o t の使い方

①グラフをかく	plot " <u>ファイル名</u> " 複数の場合: plot " <u>ファイル名</u> ", " <u>ファイル名</u> ", ...
②折れ線グラフにする	set style data lines
範囲を指定する	set xrange [<u>0:100</u>] set yrange [<u>0:100</u>]
③再描画する	replot
※画像ファイルに保存する	画面上で作図したあとで: set term png set output " <u>ファイル名.png</u> " replot
g n u p l o t を終了する	quit

プログラムの基本構造

```
#include <stdio.h>
#include <math.h>    /* この行は必要な場合のみ */
int main (void)
{

/* ここに処理の内容を書く */

    return 0;
}
```

プログラミング作業の流れ

1. Emacs でプログラムを書く。
2. コンパイルする。
3. 実行する。
4. 結果を確かめる。

第1部 放物運動のシミュレーション

①滞空時間と飛距離を計算しよう

作業の手順：以下のサンプルプログラムを Emacs で作成しよう。

```
#include <stdio.h>
#include <math.h>
int main (void)
{

    float g, V, angle, theta, T, X;          /* 変数を宣言する */

    g = 9.8;                                  /* 定数を代入する */

    printf ("Velocity [m/s], angle [deg.]?\n");
    scanf ("%f,%f", &V, &angle);           /* 条件を入力する */
    theta = 3.14159 / 180.0 * angle;

    T = 2.0 * V * sin (theta) / g;
    X = V * cos (theta) * T;

    printf ("T = %9.3f, X = %9.3f\n", T, X); /* 結果を出力する */

    return 0;
}
```

メモ：

$$T = \frac{2V \sin \theta}{g}$$

$$X = (V \cos \theta)T = \frac{2V^2 \sin \theta \cos \theta}{g} = \frac{V^2 \sin 2\theta}{g}$$

②初期の状態を計算しよう

作業の手順：①で作成したプログラムをコピーして、Emacs で完成させよう。

```
#include <stdio.h>
#include <math.h>
int main (void)
{

    float g, V, angle, theta, x, z, u, w; /* 変数を宣言する */

    g = 9.8; /* 定数を代入する */

    printf ("Velocity [m/s], angle [deg.]?\n");
    scanf ("%f,%f", &V, &angle); /* 条件を入力する */
    theta = 3.14159 / 180.0 * angle;

    x = 0.0; /* 初期値を計算する */
    z = 0.0;
    u = V * cos (theta);
    w = V * sin (theta);

    printf ("%9.3f %9.3f\n", x, z); /* 結果を出力する */

    return 0;
}
```

③0.01 秒後の状態を計算しよう

作業の手順：②で作成したプログラムをコピーして、Emacs で完成させよう。

```
#include <stdio.h>
#include <math.h>
int main (void)
{

    float g, V, angle, theta, x, z, u, w,
          dxdt, dzdt, dudt, dwdt;          /* 変数を宣言する */

    g = 9.8;                               /* 定数を代入する */

    printf ("Velocity [m/s], angle [deg.]? \n");
    scanf ("%f,%f", &V, &angle);         /* 条件を入力する */
    theta = 3.14159 / 180.0 * angle;

    x = 0.0;                               /* 初期値を計算する */
    z = 0.0;
    u = V * cos (theta);
    w = V * sin (theta);

    printf ("%9.3f %9.3f \n", x, z);       /* 結果を出力する */

    dxdt = u;                              /* 時間微分を計算する */
    dzdt = w;
    dudt = 0.0;
    dwdt = - g;

    x = x + 0.01 * dxdt;                   /* 次の時刻の値を計算する */
    z = z + 0.01 * dzdt;
    u = u + 0.01 * dudt;
    w = w + 0.01 * dwdt;

    printf ("%9.3f %9.3f \n", x, z);     /* 結果を出力する */

    return 0;
}
```

④結果をファイルに書き出そう

作業の手順：③で作成したプログラムをコピーして、Emacs で完成させよう。
実行したら出力ファイルの内容を確認しよう。

```
#include <stdio.h>
#include <math.h>
int main (void)
{

    float g, V, angle, theta, x, z, u, w,
          dxdt, dzdt, dudt, dwdt; /* 変数を宣言する */
    FILE *fp;

    g = 9.8; /* 定数を代入する */

    printf ("Velocity [m/s], angle [deg.]?\n");
    scanf ("%f,%f", &V, &angle); /* 条件を入力する */
    theta = 3.14159 / 180.0 * angle;

    x = 0.0; /* 初期値を計算する */
    z = 0.0;
    u = V * cos (theta);
    w = V * sin (theta);

    fp = fopen ("outout.txt", "w"); /* ファイルを開く */
    fprintf (fp, "%9.3f %9.3f\n", x, z); /* 結果をファイルに出力する */

    dxdt = u; /* 時間微分を計算する */
    dzdt = w;
    dudt = 0.0;
    dwdt = - g;

    x = x + 0.01 * dxdt; /* 次の時刻の値を計算する */
    z = z + 0.01 * dzdt;
    u = u + 0.01 * dudt;
    w = w + 0.01 * dwdt;

    fprintf (fp, "%9.3f %9.3f\n", x, z); /* 結果をファイルに出力する */

    fclose (fp); /* ファイルを閉じる */
    return 0;
}
```

⑤10 秒後まで計算しよう

作業の手順：④で作成したプログラムをコピーして、Emacs で完成させよう。
実行したら出力ファイルの内容を gnuplot で作図しよう。

```
#include <stdio.h>
#include <math.h>
int main (void)
{

    int i;
    float g, V, angle, theta, x, z, u, w,
          dxdt, dzdt, dudt, dwdt; /* 変数を宣言する */
    FILE *fp;

    g = 9.8; /* 定数を代入する */

    printf ("Velocity [m/s], angle [deg. ]?\n");
    scanf ("%f,%f", &V, &angle); /* 条件を入力する */
    theta = 3.14159 / 180.0 * angle;

    x = 0.0; /* 初期値を計算する */
    z = 0.0;
    u = V * cos (theta);
    w = V * sin (theta);

    fp = fopen ("output.txt", "w"); /* ファイルを開く */
    fprintf (fp, "%9.3f %9.3f\n", x, z); /* 結果をファイルに出力する */

    for (i=1; i<=1000; i++) /* 同じ処理を 1000 回繰り返す */
    {

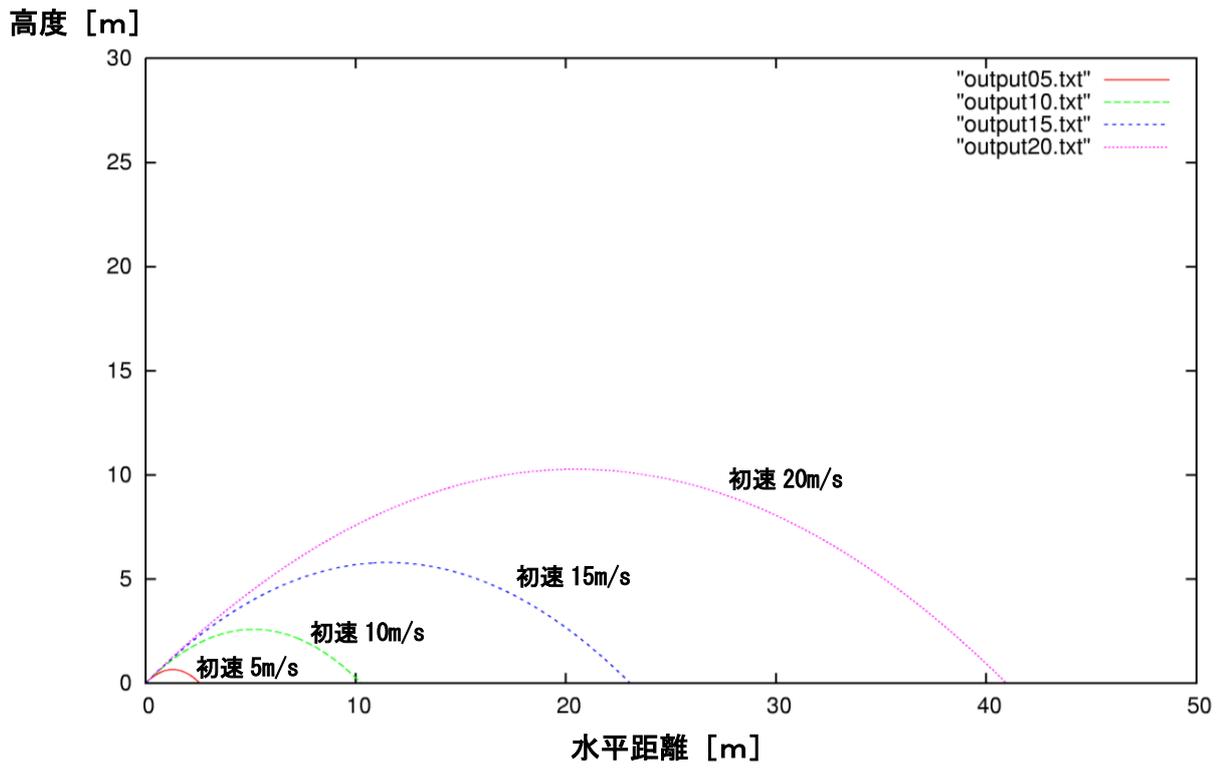
        dxdt = u; /* 時間微分を計算する */
        dzdt = w;
        dudt = 0.0;
        dwdt = - g;

        x = x + 0.01 * dxdt; /* 次の時刻の値を計算する */
        z = z + 0.01 * dzdt;
        u = u + 0.01 * dudt;
        w = w + 0.01 * dwdt;

        fprintf (fp, "%9.3f %9.3f\n", x, z); /* 結果をファイルに出力する */

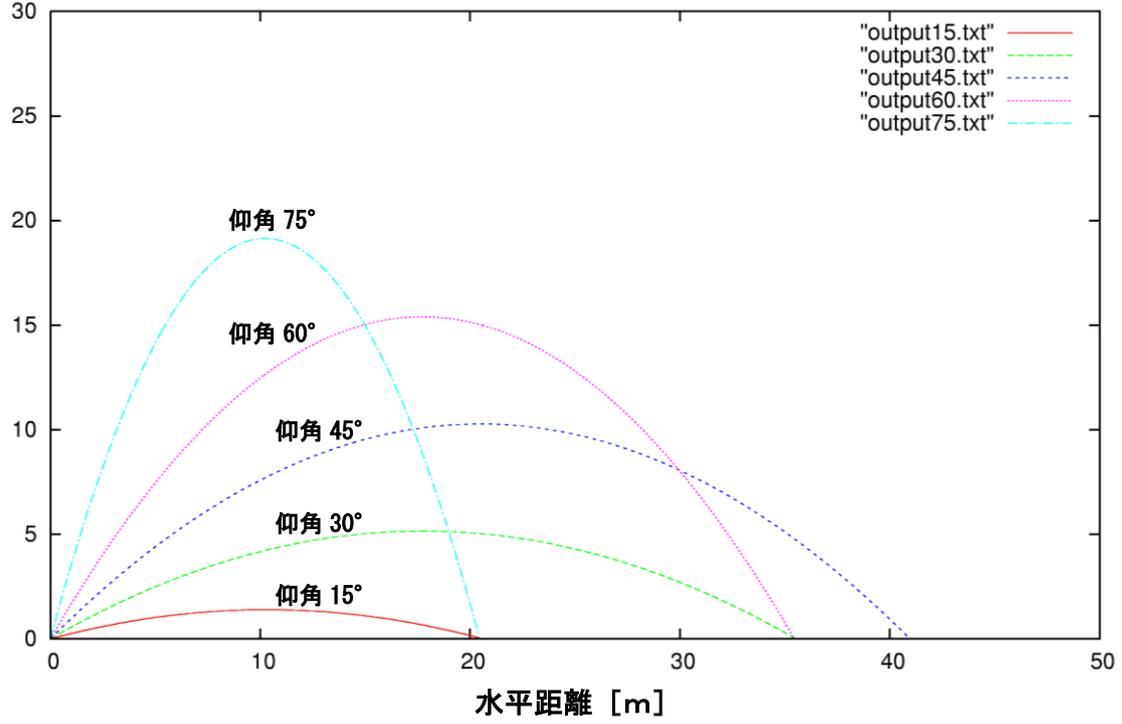
    }

    fclose (fp); /* ファイルを閉じる */
    return 0;
}
```



計算結果の例（仰角 45° の場合）

高度 [m]



計算結果の例（初速 20m/s の場合）