二次抵抗を含む弾道のオイラー法解法メモ

綴木 馴（つづるぎ じゅん）

November 9, 2025

目的

砲弾を仰角 θ で発射し，x **方向・** y **方向の両方**に空気抵抗（速度の二乗に比例）が作用する場合の運動方程式を示し，陽的オイラー法での数値解法手順をまとめる．

前提と記法

- 質点近似，地球は平坦，風は無視．
- 質量を m，重力加速度を g，位置を $\boldsymbol{r} = (x, y)$，速度を $\boldsymbol{v} = (v_x, v_y)$ とする．
- 抗力係数 $k = \frac{1}{2}\rho C_\mathrm{d} A$（$\rho$：空気密度，$C_\mathrm{d}$：抗力係数，$A$：抗力基準面積）とし，抗力は $\boldsymbol{F}_\mathrm{D} = -k\|\boldsymbol{v}\|\,\boldsymbol{v}$ とおく．
- 初期条件：$x(0) = 0,\ y(0) = 0,\ v_x(0) = v_0 \cos\theta,\ v_y(0) = v_0 \sin\theta.$

運動方程式（ニュートンの第二法則）

重力 $\boldsymbol{F}_\mathrm{G} = (0, -mg)$ と抗力 $\boldsymbol{F}_\mathrm{D}$ のみを考えると，

$$m\frac{\mathrm{d}\boldsymbol{v}}{\mathrm{d}t} = \boldsymbol{F}_\mathrm{G} + \boldsymbol{F}_\mathrm{D} = (0, -mg) - k\|\boldsymbol{v}\|\,\boldsymbol{v}, \tag{1}$$

$$\frac{\mathrm{d}\boldsymbol{r}}{\mathrm{d}t} = \boldsymbol{v}. \tag{2}$$

成分表示すると，x **方向にも** y **方向にも**二乗抵抗が入る：

$$m\frac{\mathrm{d}v_x}{\mathrm{d}t} = -k\,v\,v_x, \tag{3}$$

$$m\frac{\mathrm{d}v_y}{\mathrm{d}t} = -mg - k\,v\,v_y, \tag{4}$$

$$\frac{\mathrm{d}x}{\mathrm{d}t} = v_x, \quad \frac{\mathrm{d}y}{\mathrm{d}t} = v_y, \tag{5}$$

ただし $v = \|\boldsymbol{v}\| = \sqrt{v_x^2 + v_y^2}$.
表記の理由：抗力は速度ベクトルと逆向きに働き，その大きさは速度の大きさ v に比例する．したがって成分表示では x 成分は $-k\,v\,v_x$ となる．これは「抗力は速度ベクトルの方向に沿って作用する」という物理的性質を保つ表現である．これを ${v_x}^2$ と書くと，符号情報が失われ，あたかも x 軸方向の大きさのみに依存するかのような誤解を与えるため不適切である（$v\,v_x = \sqrt{v_x^2 + v_y^2}\,v_x$）．

備考　よくある誤りは，y 方向だけに抗力を入れること．(3) の右辺にも $-k\,v\,v_x$ が必ず現れる．

離散化：陽的オイラー法

時間刻み Δt，離散時刻 $t_n = n\Delta t$ に対し，

$$x_{n+1} = x_n + \Delta t\, v_{x,n}, \tag{6}$$
$$y_{n+1} = y_n + \Delta t\, v_{y,n}, \tag{7}$$
$$v_{x,n+1} = v_{x,n} + \Delta t \left(-\frac{k}{m}\, v_n\, v_{x,n} \right), \tag{8}$$
$$v_{y,n+1} = v_{y,n} + \Delta t \left(-g - \frac{k}{m}\, v_n\, v_{y,n} \right), \tag{9}$$

ここで $v_n = \sqrt{v_{x,n}^2 + v_{y,n}^2}$．初期値は

$$x_0 = 0,\ y_0 = 0,\ v_{x,0} = v_0 \cos\theta,\ v_{y,0} = v_0 \sin\theta. \tag{10}$$

着弾条件　y が 0 を下回った最初のステップで停止し，線形補間で到達距離 R を補正すると精度が向上する．

アルゴリズム（C 言語）

```
/* Euler method for projectile with quadratic drag in x and y.
   Model:
     dvx/dt = -(k/m) * v * vx
     dvy/dt = -g      - (k/m) * v * vy
     dx/dt  =  vx
     dy/dt  =  vy
   where v = sqrt(vx*vx + vy*vy).
*/
#include <stdio.h>
#include <math.h>

int main(void) {
    /* parameters */
    const double m = 1.0;          /* kg */
    const double g = 9.80665;      /* m/s^2 */
    const double k = 0.05;         /* kg/m = 0.5*rho*Cd*A */
    const double v0 = 200.0;       /* m/s */
    const double theta = 30.0 * M_PI / 180.0; /* rad */
    const double dt = 0.001;       /* s */

    /* initial conditions */
    double t = 0.0;
    double x = 0.0, y = 0.0;
    double vx = v0 * cos(theta);
    double vy = v0 * sin(theta);
```

```
    /* for range interpolation */
    double prev_x = x, prev_y = y;

    printf("# t x y vx vy\n");
    printf("%.6f %.8f %.8f %.8f %.8f\n", t, x, y, vx, vy);

    while (y >= 0.0) {
        prev_x = x; prev_y = y;

        /* speed magnitude */
        double v = sqrt(vx*vx + vy*vy);

        /* Euler updates (explicit) */
        vx = vx + dt * (-(k/m) * v * vx);
        vy = vy + dt * (-g - (k/m) * v * vy);
        x  = x  + dt * vx;
        y  = y  + dt * vy;
        t  = t  + dt;

        printf("%.6f %.8f %.8f %.8f %.8f\n", t, x, y, vx, vy);
    }

    /* Linear interpolation to estimate range at y=0 */
    double R = x;
    if (y < 0.0) {
        double alpha = prev_y / (prev_y - y);     /* in (0,1) */
        R = prev_x + alpha * (x - prev_x);
    }
    fprintf(stderr, "Estimated range (y=0): %.8f m\n", R);
    return 0;
}
```

記号と次元

記号	意味（SI 単位）
m	質量（kg）
g	重力加速度（$\mathrm{m\,s^{-2}}$）
k	抗力係数 $\frac{1}{2}\rho C_\mathrm{d} A$（$\mathrm{kg\,m^{-1}}$）
$\boldsymbol{v} = (v_x, v_y)$	速度（$\mathrm{m\,s^{-1}}$）
$\boldsymbol{r} = (x, y)$	位置（m）
$v = \lVert\boldsymbol{v}\rVert$	速度の大きさ（$\mathrm{m\,s^{-1}}$）
θ	発射仰角（rad）
v_0	発射速度（$\mathrm{m\,s^{-1}}$）
Δt	時間刻み（s）

数値安定性の注意

オイラー法は一次精度で数値拡散が大きい．Δt を十分小さく取り（例：飛行時間の 10^4 分割程度），必要に応じて改良オイラー法や 4 次 Runge–Kutta 法を用いると良い．

拡張の例（任意）

- 風速 $\boldsymbol{u}$ があるときは相対速度 $\boldsymbol{v}-\boldsymbol{u}$ のノルムを用いて $-k\|\boldsymbol{v}-\boldsymbol{u}\|(\boldsymbol{v}-\boldsymbol{u})$ に置換する．
- 密度の高度依存 $\rho(y)$ を入れると $k \to k(y)$ となる．