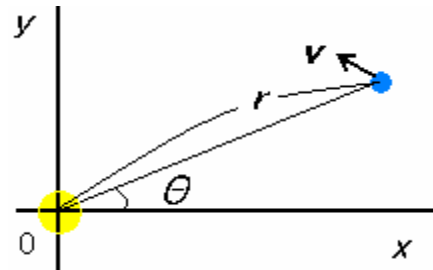


2006年5月18日 (木) 実施

中心力による運動 (2次元)

太陽の周りを回る地球の運動を考える。両者の間には万有引力が働く。運動は、厳密には太陽と地球との重心の周りの回転運動となるが、太陽の質量は地球の質量に比べて大きいので、近似的には太陽を中心とする中心力 $\vec{f} = -G \frac{Mm}{r^2} \frac{\vec{r}}{r}$ を受けた回転運動



となる。(G : 万有引力定数, M : 太陽の質量, m : 地球の質量)

$x-y$ 平面で考えると, $r^2 = x^2 + y^2$ となる。速度の各成分を $\frac{dx(t)}{dt} = v_x(t)$, $\frac{dy(t)}{dt} = v_y(t)$

と表し, 時刻 $t=0$ での速度を $v_x(0) = 0$, $v_y(0) = v_0$ とする。地球に働く力の成分は,

$$f_x = m \frac{dv_x(t)}{dt} = -m \frac{GM}{x^2 + y^2} \frac{x}{\sqrt{x^2 + y^2}}, \quad f_y = m \frac{dv_y(t)}{dt} = -m \frac{GM}{x^2 + y^2} \frac{y}{\sqrt{x^2 + y^2}} \text{ となる。}$$

例題

ルンゲクッタ法を用いて地球の公転運動 (太陽の周りの回転運動) を解く C プログラムを作成し, 翻訳編集して実行する。また, ファイルに結果を書き出し, GNU PLOT でグラフ化する。ただし, $G = 6.673 \times 10^{-11} \text{ Nm}^2\text{kg}^{-2}$, $M = 1.9891 \times 10^{30} \text{ kg}$, $x(0) = 1.5 \times 10^{11} \text{ m}$, $y(0) = 0 \text{ m}$, $v_0 = 3.0 \times 10^4 \text{ ms}^{-1}$ とする。

ex4-1.c

```
#include <stdio.h>
#include <math.h>

#define G 6.673E-11
#define M 1.9891E30
#define TMAX 3.1536E7

double fx(double, double);
double fy(double, double);
double fvx(double, double, double);
double fvy(double, double, double);

int main(void)
{
```

自然科学シミュレーション ノート

```
double x, y, vx, vy, t, dt;
double k0x, k1x, k2x, k3x, k0vx, k1vx, k2vx, k3vx;
double k0y, k1y, k2y, k3y, k0vy, k1vy, k2vy, k3vy;
FILE *output;

t=0.0;
dt=0.01*TMAX;
x=1.5E11;
y=0.0;
vx=0.0;
vy=3.0E4;

output=fopen("solar.data", "w");

fprintf(output, "%E %E %E\n", x, y, t);

for (t=0.0; t<TMAX; t+=dt)
{
    k0x=dt*fx(t, vx);
    k0y=dt*fy(t, vy);
    k0vx=dt*fvx(t, x, y);
    k0vy=dt*fvy(t, x, y);
    k1x=dt*fx(t+dt/2.0, vx+k0vx/2.0);
    k1y=dt*fy(t+dt/2.0, vy+k0vy/2.0);
    k1vx=dt*fvx(t+dt/2.0, x+k0x/2.0, y+k0y/2.0);
    k1vy=dt*fvy(t+dt/2.0, x+k0x/2.0, y+k0y/2.0);
    k2x=dt*fx(t+dt/2.0, vx+k1vx/2.0);
    k2y=dt*fy(t+dt/2.0, vy+k1vy/2.0);
    k2vx=dt*fvx(t+dt/2.0, x+k1x/2.0, y+k1y/2.0);
    k2vy=dt*fvy(t+dt/2.0, x+k1x/2.0, y+k1y/2.0);
    k3x=dt*fx(t+dt, vx+k2vx);
    k3y=dt*fy(t+dt, vy+k2vy);
    k3vx=dt*fvx(t+dt, x+k2x, y+k2y);
    k3vy=dt*fvy(t+dt, x+k2x, y+k2y);
    x=x+(k0x+2.0*k1x+2.0*k2x+k3x)/6.0;
    y=y+(k0y+2.0*k1y+2.0*k2y+k3y)/6.0;
    vx=vx+(k0vx+2.0*k1vx+2.0*k2vx+k3vx)/6.0;
    vy=vy+(k0vy+2.0*k1vy+2.0*k2vy+k3vy)/6.0;

    fprintf(output, "%E %E %E\n", x, y, t+dt);
}

fclose(output);
```

```
    return 0;
}

double fx(double t, double vx)
{
    return vx;
}

double fy(double t, double vy)
{
    return vy;
}

double fvx(double t, double x, double y)
{
    return -G*M/(x*x+y*y)*x/sqrt(x*x+y*y);
}

double fvy(double t, double x, double y)
{
    return -G*M/(x*x+y*y)*y/sqrt(x*x+y*y);
}
```

* このプログラムで書き出した solar.data を GNUPLOT で描画するには、3次元プロットに対応した splot を用いて、次のように範囲指定を行うと見やすい。

```
splot [-1.6e11:1.6e11] [-1.6e11:1.6e11] [0:3.2e7] 'solar.data' with
linespoints
```

なお、ここで描画される曲線は時間の流れを上向きに取った地球の軌跡を表す「世界線」と呼ばれるものである。

プログラムを書き換えて、fprintf で x 及び y のみを書き出せば、x-y 平面上の地球の軌跡が得られる。