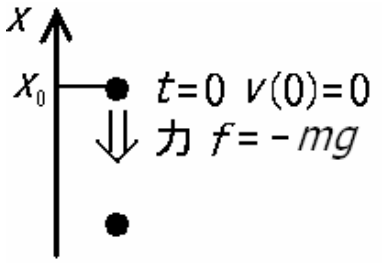
2006 年 4 月 27 日（木）実施

自由落下



x 軸を鉛直上向きに取る。時刻 $t=0$ で速度 $v(0)=0$，位置座標 $x(0)=x_0$ とする。物体に働く力は，$\dfrac{dx(t)}{dt}=v(t)$ を用いて，$f=m\dfrac{dv(t)}{dt}=-mg$ となる。（地表近くの近似）

・微分方程式 $\dfrac{dv(t)}{dt}=-g$ を数値的に解く（オイラー法）

十分小さな差分 Δt を用いると，$\dfrac{dv(t)}{dt}\cong\dfrac{v(t+\Delta t)-v(t)}{\Delta t}$ と表せる。この式から，

$v(t+\Delta t)\cong v(t)+\dfrac{dv(t)}{dt}\Delta t=v(t)-g\ \Delta t$ の時間発展を追えばよい。

更に，$\dfrac{dx(t)}{dt}=v(t)$ を数値的に解くには，$x(t+\Delta t)\cong x(t)+v(t)\Delta t$ の時間発展を追えばよいが，この式で単純に $t=0$ とすると，$x(\Delta t)\cong x_0$ となってしまう。これは，$v(t)\cong v(t+\Delta t)+g\ \Delta t$ には $(\Delta t)^2$ に比例した誤差があるからである。$v(t)$ を $v(t+\Delta t)$ で表した表式を使って，$(\Delta t)^2$ 項を落とすと，$x(t+\Delta t)\cong x(t)+v(t+\Delta t)\Delta t$ となる。この場合には，$t=0$ を代入すると $x(\Delta t)\cong x_0+v(\Delta t)\Delta t=x_0-g\ (\Delta t)^2$ となる。実は，これも解析的に微分方程式を解いた場合とは一致しない。$v(t)$ と $v(t+\Delta t)$ の平均値を用いた修正オイラー法を用いれば，$x(t+\Delta t)\cong x(t)+\dfrac{v(t+\Delta t)+v(t)}{2}\Delta t$，$x(\Delta t)\cong x_0-\dfrac{1}{2}g\ (\Delta t)^2$ となり，精度がよくなる。（$x(\Delta t)$ については解析的に微分方程式を解いた場合と一致）

例題

オイラー法を用いて自由落下する物体の速度を求める 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，GNUPLOT でグラフ化する。

ex2-1.c

```
#include <stdio.h>

int main(void)
{
```

```
   double dt,t,v;
   int i;
   dt=0.01;
   t=0.0;
   v=0.0;
   printf(”時刻= %f, 速度= %f¥n”, t, v);
   for (i=0;i<10;i++)
   {
      t=t+dt;
      v=v-9.8*dt;
      printf(”時刻= %f, 速度= %f¥n”, t, v);
   }
   return 0;
}
```

ex2-2.c

```
#include <stdio.h>
int main(void)
{
   double dt,t,v;
   int i;
   FILE *output;
   dt=0.01;
   t=0.0;
   v=0.0;
   output=fopen(”sokudo.data”,”w”);
   fprintf(output,”%f %f¥n”, t, v);
   for (i=0;i<10;i++)
   {
      t=t+dt;
      v=v-9.8*dt;
      fprintf(output,”%f %f¥n”, t, v);
   }
   fclose(output);
   return 0;
}
```