２０２３年度前期　数値解析　第１４回講義ノート

２０２３年７月２０日　三浦憲二郎

スライド１　講義アウトライン

- 常微分方程式
  - １段階法と多段階法
  - 「　　」法
  - 「　　」法
  - 「　　」法

スライド２　１段階法と多段階法　p.154　復習

- f(x,y)は $a \leqq x \leqq b$, $y \in \Re$で定義された２変数関数とする．１階の常微分方程式　y’=f(x,y) を考える．このとき，任意の $x \in [a,b]$に対して y’=f(x,y(x))となるような C^1級の関数 y=y(x)を常微分方程式の一般解または解という．y_0が与えられたとき，$y(a)=y_0$ を満たすような解を求める問題を常微分方程式の初期値問題という．

$$\begin{aligned} y' &= f(x,y) \\ y(a) &= y_0 \end{aligned}$$

- 数値解法
- １段階法
- 多段階法

スライド３　１段階法と多段階法　p.154

- 区間[a,b]を等間隔 $a=x_0<x_1<x_2<\cdots<x_n=b$, $h=(b-a)/n$ に分割，$\phi(x,y)$ は与えられた関数，関数値$y(x_i)$の数値解を Y_i とする．
- 「　　」法

$$\begin{aligned} Y_0 &= y_0 \\ Y_{i+1} &= Y_i + h\phi(x_i, Y_i),\ i = 0,1,2,\cdots,n-1 \end{aligned}$$

- 「　　」法　　定数α_j(j=0,1,…,k-1)

$$\begin{aligned} Y_0 &= y_0 \\ Y_{i+1} &= \alpha_0 Y_i + \alpha_1 Y_{i-1} + \cdots + \alpha_{k-1} Y_{i-k+1} \\ &\quad + h\phi(x_i, x_{i-1}, \cdots, x_{i-k+1}, Y_i, Y_{i-1}, \cdots, Y_{i-k+1}, h),\ i = 0,1,2,\cdots,n-1 \end{aligned}$$

右辺に Y_{i+1} が含まれている公式：「　　」公式

いない公式：「　　」公式

スライド４　１段階法と多段階法　p.155

- １段階法

$$y_{i+1} = y_i + h\phi(x_i, y_i) + O(h^{p+1})$$

  - 次数 p の「　　」を持つ，または p 次の「　　」をもつという．
  - 「　　」誤差

$$\tau_i = \frac{y_{i+1} - y_i}{h} - \phi(x_i, y_i)$$

- 多段階法

$$\begin{aligned} y_{i+1} &= \alpha_0 y_i + \alpha_1 y_{i-1} + \cdots + \alpha_{k-1} y_{i-k+1} \\ &\quad + h\phi(x_i, x_{i-1}, \cdots, x_{i-k+1}, Y_i, Y_{i-1}, \cdots, Y_{i-k+1}, h) + O(h^{p+1}) \end{aligned}$$

  - 「　　」p の精度を持つ，または p「　　」の精度をもつという．
  - 「　　」誤差

$$\begin{aligned} \tau_{i+1} &= \frac{y_{i-1} - (\alpha_0 y_i + \alpha_1 y_{i-1} + \cdots + \alpha_{k-1} y_{i-k+1})}{h} \\ &\quad - \phi(x_i, x_{i-1}, \cdots, x_{i-k+1}, y_i, y_{i-1}, \cdots, y_{i-k+1}, h) \end{aligned}$$

スライド５　オイラー法　　p.155

- [a,b]を n 等分し，各分点 x_i を

$$h = \frac{b-a}{n},\ x_0 = a,\ x_i = a + ih\ (i = 1, 2, \cdots, n)$$

  h：「　　」幅

  $y_i = y(x_i)$ とすると，「　　」の定理より

$$\begin{aligned} y_{i+1} &= y(x_i) + hy'(x_i) + \frac{h^2}{2} y''(x_i + \theta h),\ 0 < \theta < 1 \\ &= y_i + hf(x_i, y_i) + O(h^2) \end{aligned}$$

- オイラー法　（「　　」次の精度を持つ「　　」段階法）

$$\begin{aligned} Y_0 &= y_0 \\ Y_{i+1} &= Y_i + hf(x_i, Y_i),\ i = 0, 1, 2, \cdots, n-1 \end{aligned}$$

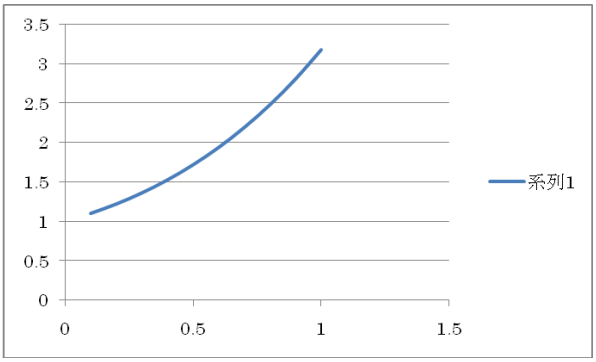
スライド６　オイラー法：プログラム　p.157

```
#include <stdio.h>
/*  関数の定義  */
double func(double x, double y);
/*  オイラー法  */
void euler(double x, double y, double a, double b, int n,
                                    double (*f)(double, double) );
int main(void)
{
   int n;
   printf("分割数を入力してください--->");
   scanf("%d",&n);
   euler( 0.0, 1.0, 0.0, 1.0, n, func );
   return 0;
```

実行結果

```
分割数を入力してください--- > 10
x=0.1     y=1.1
x=0.2     y=1.22
x=0.3     y=1.362
x=0.4     y=1.5282
x=0.5     y=1.72102
x=0.6     y=1.943122
x=0.7     y=2.197434
x=0.8     y=2.487178
x=0.9     y=2.815895
x=1       y=3.187485
```

```
}
/* オイラー法 */
void euler(double x, double y, double a, double b, int n, double (*f)(double, double) )
{
   double h;
   int i;
   h = (b-a)/n;   /* 刻み幅 */
   for ( i = 0 ; i < n ; i++) {
      y = y + h * (*f)(x,y);
      x += h;
      printf("x=%f ¥t y=%f ¥n", x, y );
   }
}
/* 関数の定義 */
double func(double x, double y)
{
   return( x + y );
}
```



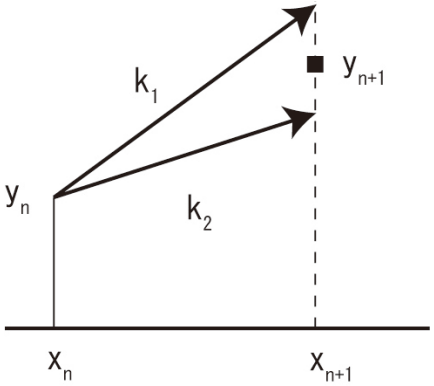
スライド７　ホイン法　p.158

次数「　　」の「　　」段階法　　ホイン法　（Heun method）

$$Y_{i+1} = Y_i + h\phi(x_i, Y_i),\ i = 0, 1, 2, \cdots, n-1$$
$$\phi(x_i, Y_i) = \frac{1}{2}(f(x_i, Y_i) + f(x_{i+1}, Y_i + hf(x_i, Y_i)))$$

ホイン法アルゴリズム

Input x, Y, a, b, n
$h \leftarrow (b-a)/n$
for $i = 0, 1, 2, \cdots, n-1$
　$k_1 \leftarrow f(x, Y)$
　$k_2 \leftarrow f(x+h, Y+hk_1)$
　$Y \leftarrow Y + \frac{h}{2}(k_1 + k_2)$
　$x \leftarrow x + h$
　Output Y
end for



スライド８　ホイン法：プログラム　p.159

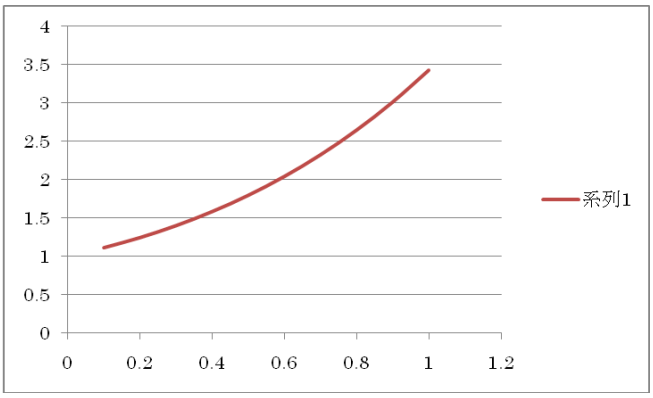
```
#include <stdio.h>
/* 関数の定義 */
double func(double x, double y);
/* ホイン法 */
void heun(double x, double y, double a, double b, int n,
                                    double (*f)(double, double) );
int main(void)
{
   int n;
   printf("分割数を入力してください--->");
   scanf("%d",&n);
   heun( 0.0, 1.0, 0.0, 1.0, n, func );
   return 0;
}
```

実行結果

分割数を入力してください--- > 10

x=0.1	y=1.11
x=0.2	y=1.24205
x=0.3	y=1.398465
x=0.4	y=1.581804
x=0.5	y=1.794894
x=0.6	y=2.040857
x=0.7	y=2.323147
x=0.8	y=2.645578
x=0.9	y=3.012364
x=1	y=3.428162

```
/* ホイン法 */
void heun(double x, double y, double a, double b, int n, double (*f)(double, double) )
{
   double k1, k2, h;
   int i;
   h = (b-a)/n;
   for ( i = 0 ; i < n ; i++){
      k1 = f(x,y); k2 = f(x+h, y+h*k1);
      y = y + h/2.0 * ( k1 + k2 );
      x += h;
      printf("x=%f ¥t y=%f ¥n", x, y );
   }
}
/* 関数の定義 */
double func(double x, double y)
{
   return( x + y );
}
```



スライド９　ルンゲ・クッタ法　　p.160

- １段階法でもっともよく使われる．　次数「　　」の「　　」法(Runge-Kutta)

$$
\begin{aligned}
Y_0 &= y_0 \\
Y_{i+1} &= Y_i + h\phi(x_i, Y_i) \\
\phi(x_i, Y_i) &= \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \\
\text{where} \quad & k_1 = f(x_i, Y_i), \\
& k_2 = f(x_i + \frac{h}{2}, Y_i + \frac{h}{2}k_1), \\
& k_3 = f(x_i + \frac{h}{2}, Y_i + \frac{h}{2}k_2), \\
& k_4 = f(x_i + h, Y_i + hk_3)
\end{aligned}
$$

$$y_{i+1} = y_i + h\phi(x_i, y_i) + O(h^5)$$

スライド１０　ルンゲ・クッタ法：プログラム　　p.161

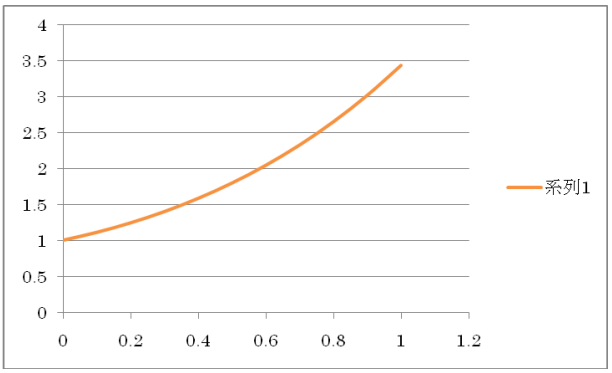
```
#include <stdio.h>
#include <stdlib.h>
double *dvector(long i, long j);          /* ベクトル領域の確保 */
void free_dvector(double *a, long i); /* 領域の解放 */
double func(double x, double y);          /* 関数の定義 */
/* ルンゲ・クッタ法 */
double *rk4( double y0, double *y, double a, double b, int n,
                                    double (*f)(double, double) );
int main(void)
{
   double *y, h, a=0.0, b=1.0, y0=1.0 ;
   int i, n;
   printf("分割数を入力してください--->");
   scanf("%d",&n);
   y = dvector( 0, n );                    /* 領域の確保 */
   y = rk4( y0, y, a, b, n, func );   /* ルンゲ・クッタ法 */
   /* 結果の表示 */
   h = (b-a)/n ;                  /* 刻み幅 */
   for ( i = 0 ; i <= n ; i++) {
```

実行結果

分割数を入力してください--- > 10

x=0	y=1
x=0.1	y=1.110342
x=0.2	y=1.242805
x=0.3	y=1.399717
x=0.4	y=1.583648
x=0.5	y=1.797441
x=0.6	y=2.044236
x=0.7	y=2.327503
x=0.8	y=2.651079
x=0.9	y=3.019203
x=1	y=3.436559

```
    printf("x=%f ¥t y=%f ¥n", a+i*h, y[i] );
  }
  free_dvector( y, 0 ); /*  領域の解放  */
  return 0;
}
/*  ルンゲ・クッタ法  */
double *rk4( double y0, double *y, double a, double b, int n,
                                        double (*f)(double, double) )
{
  double k1, k2, k3, k4, h, x;
  int i;
  h = (b-a)/n;
  /*  初期値の設定  */
  y[0] = y0; x = a;
  /*  ルンゲ・クッタ法  */
  for ( i = 0 ; i < n ; i++) {
    k1 = f(x,y[i]); k2 = f(x+h/2.0, y[i]+h*k1/2.0);
    k3 = f(x+h/2.0, y[i]+h*k2/2.0);
    k4 = f(x+h, y[i]+h*k3);
    y[i+1] = y[i] + h/6.0 * ( k1 + 2.0*k2 + 2.0*k3 + k4 );
    x += h;
  }
  return y;
}
```



スライド１１　ルンゲ・クッタ型公式の導出

- 微分積分学入門第二課，一松　信，近代科学社　p.182
- f(x,y)が十分に滑らかとすると，f’(x,y)の高階導関数は，合成関数の微分の公式により

$$
\begin{aligned}
y'' &= f_x + f f_y, \\
y''' &= f_{xx} + 2f f_{xy} + f^2 f_{yy} + f_y(f_x + f f_y), \\
y'''' &= f_{xxx} + 3f f_{xxy} + 3f^2 f_{xyy} + f^3 f_{yyy} + 3(f_{xy} + f f_{yy})(f_x + f f_y) \\
&\quad + f_y(f_{xx} + 2f f_{xy} + f^2 f_{yy}) + f_y^2(f_x + f f_y)
\end{aligned}
$$

f_x, f_y は x, y に関する偏微分

$$k_i = hf(x + h\beta_i, y + \sum_{j=1}^{p} \alpha_{ij} k_j)$$

$$
\begin{aligned}
k_i &= hf + h^2\beta_i(f_x + f f_y) + h^3 \sum_{j=1}^{p} \alpha_{ij}\beta_j f_y(f_x + f f_y) \\
&\quad + \frac{h^3}{2}\beta_i^2(f_{xx} + 2f f_{xy} + f^2 f_{yy}) \\
&\quad + h^4\beta_i \sum_{j=1}^{p} \alpha_{ij}\beta_j(f_x + f f_y)(f_{xy} + f f_{yy}) \\
&\quad + \frac{h^4}{2} \sum_{j=1}^{p} \alpha_{ij}\beta_j^2 f_y(f_{xx} + 2f f_{xy} + f^2 f_{xy}) \\
&\quad + h^4 \sum_{j=1}^{p}\sum_{l=1}^{p} \alpha_{ij}\alpha_{jl}\beta_l f_y^2(f_x + f f_y) + \frac{h^4}{6}\beta_i^3(f_{xxx} + 3f f_{xxy} + 3f^2 f_{xyy} + f^3 f_{yyy}) + O(h^5)
\end{aligned}
$$

- $\Sigma\, \omega_i k_i$ を真の解のテイラー展開の１次以上４次以下の項

$$hy' + \frac{h^2}{2}y'' + \frac{h^3}{6}y''' + \frac{h^4}{24}y''''$$

$$
\begin{aligned}
h \quad & \sum_{i=1}^{p} \omega_i = 1, \\
h^2 \quad & \sum_{i=1}^{p} \omega_i \beta_i = \frac{1}{2}, \\
h^3 \quad & \sum_{i=1}^{p} \omega_i \beta_i^2 = \frac{2}{6} = \frac{1}{3}, \\
& \sum_{i=1}^{p} \omega_i \alpha_{ij} \beta_i^2 = \frac{1}{6}, \\
h^4 \quad & \sum_{i=1}^{p} \omega_i \beta_i^3 = \frac{6}{24} = \frac{1}{4}, \\
& \sum_{i=1}^{p} \sum_{j=1}^{p} \omega_i \alpha_{ij} \beta_j^2 = \frac{2}{24} = \frac{1}{12}, \\
& \sum_{i=1}^{p} \sum_{j=1}^{p} \omega_i \beta_i \alpha_{ij} \beta_j = \frac{3}{24} = \frac{1}{8}, \\
& \sum_{i=1}^{p} \sum_{j=1}^{p} \sum_{l}^{p} \omega_i \alpha_{ij} \alpha_{jl} \beta_i = \frac{1}{24}.
\end{aligned}
$$

$$
\begin{aligned}
k_i &= hf(x_n + \beta_i h, y_n + \sum_{j=1}^{p} \alpha_{ij} k_j), \ (i = 1, 2, \cdots, p) \\
y_{n+1} &= y_n + \Delta y_n, \ \Delta y_n = \sum_{i=1}^{p} \omega_i k_i.
\end{aligned}
$$

$$
\begin{array}{ccccccc|c}
0 & 0 & 0 & \cdots & 0 & 0 & 0 \\
\alpha_{21} & 0 & 0 & \cdots & 0 & 0 & \beta_2 \\
\alpha_{31} & \alpha_{32} & 0 & \cdots & 0 & 0 & \beta_3 \\
 & & & \cdots & & \vdots & \\
\alpha_{p1} & \alpha_{p2} & \alpha_{p3} & \cdots & \alpha_{p,p-1} & 0 & \beta_p \\
\hline
\omega_1 & \omega_2 & \omega_3 & \cdots & \omega_{p-1} & \omega_p &
\end{array}
$$

段：f を計算する回数

位：一致する次数

1段1位　オイラー法

$$
\begin{array}{cc|c}
0 & 0 & 0 \\
\hline
1 & & 1
\end{array}
\qquad y_{n+1} = y_n + hf(x_n, y_n)
$$

2段1位　ホイン法

$$
\begin{array}{cc|c}
0 & 0 & 0 \\
1 & 0 & 1 \\
\hline
1/2 & 1/2 &
\end{array}
\qquad y_{n+1} = y_n + \frac{h}{2}(f(x_n, y_n) + f(x_{n+1}, y_n + hf(x_n, y_n)))
$$

4段4位　ルンゲ・クッタ法

$$
\begin{array}{cccc|c}
0 & 0 & 0 & 0 & 0 \\
1/2 & 0 & 0 & 0 & \\
0 & 1/2 & 0 & 1/2 & \\
0 & 0 & 1 & 0 & 1 \\
\hline
1/6 & 1/3 & 1/3 & 1/6 &
\end{array}
$$