

Chapter 1

常微分方程式

1.1 1階微分方程式の初期値問題

$$\begin{aligned}\frac{dx}{dt} &= f(t, x) \\ y(0) &= a \quad (a : \text{定数})\end{aligned}\tag{1.1}$$

を考えます。もっとも基本になるのはオイラー法で以下の漸化式で (t_0, x_0) から始めて $(t_1, x_1), (t_2, x_2), \dots$ の順に解の近似値を求めます。ただし、 h は t の刻み幅で、計算で具体的な数値を指定します。

$$\begin{aligned}s_1 &= f(t_n, x_n) \\ x_{n+1} &= x_n + h s_1 \\ t_{n+1} &= t_n + h\end{aligned}\tag{1.2}$$

オイラー法の VBA によるプログラムが program 1-1 です。関数文を使っているので関数文にある F を変化させることによっていろいろな微分方程式を解くことができます。この例では $F = x$ としてあり、 t を含んでいませんが文法的には問題はありません。また $a = 1$ という初期条件で $h = 0.1$ にとっていますが、この部分も問題によって変化させます。なお、この場合は厳密解 $x = \exp(t)$ をもつのでそれとの比較もおこなっています（通常は厳密解がわからないのでこのような比較はできません）。Fig.1.1 には計算結果を表と散布図で示していますが、 t が大きくなるにつれて差が目立つようになります。

program 1-1

```
' 1階常微分方程式 - オイラー法
Sub Euler()
    Cells.Clear
    H = 0.1
    Tmax = 3#
```

```

Nmax = Tmax / H
T = 0#
X = 1#
For N = 0 To Nmax
  Cells(N + 1, 1) = T
  Cells(N + 1, 2) = X
  Cells(N + 1, 3) = Exp(T)
  S1 = F(T, X)
  X = X + S1 * H
  T = T + H
Next N
End Sub

Function F(T, X)
  F = X
End Function

```

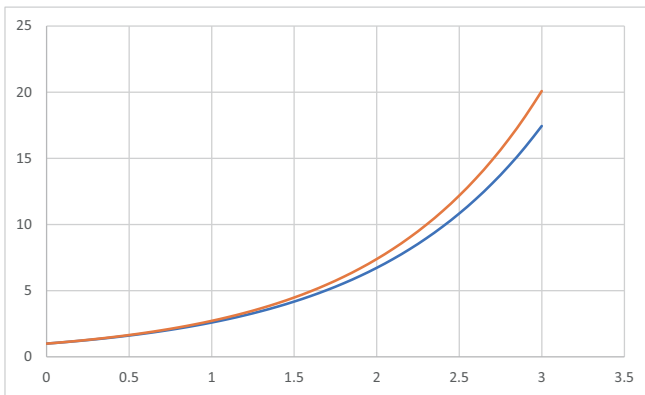


Fig.1.1

一般の BASIC ではサポートされていない VBA 独自の命令として、Cells.clear という命令を使っていますが、これは計算初期にエクセルの表を値の入っていない初期状態にするものです。これをつけないと以前あった表を上書きされますので、どこまでが新しい結果であるかわからなくなるので、書いておくと便利です。他に Cells(I,J)=A という形の命令も使っています。これは A に対応する数値をセルの I 行、J 列に書き込むという命令です。I や J をループでまわすことにより表が作成できます。数値計算では配列を多用しますが、Cells も 2 次元配列とみなすことができます。なお、VBA で配列を用いるときはあらかじめ宣言しておく必要があります (Cells は組み込まれているので宣言はしません)。注意すべきことは VBA における配列の要素は 0

からはじまるということです。たとえば

DIM B(10)

という命令を書いたときは、配列 B に対して要素は $B(0), B(1), \dots, B(10)$ となります。2次元以上の配列に対しても同じです。一方 Cells(0,0) とするとエラーになります。表には0行目や0列目がありません。

オイラー法より精度のよい方法に**ルンゲ・クッタ法**があります。ルンゲ・クッタ法にもいろいろ種類がありますが、ここでは手軽な2次のルンゲ・クッタ法と精度がよいためよく使われる4次のルンゲ・クッタ法(1/6公式)に対し、それらを式(1.1)に適応したプログラムを program 1-2 と program 1-3 にのせておきます。これらはそれぞれ式(1.1)を解くための以下のアルゴリズムがもとになっています。

$$\begin{aligned} s_1 &= f(t_n, x_n) \\ s_2 &= f(t_n + h, x_n + hs_1) \\ x_{n+1} &= x_n + h(s_1 + s_2)/2 \\ t_{n+1} &= t_n + h \end{aligned} \tag{1.3}$$

$$\begin{aligned} s_1 &= f(t_n, x_n) \\ s_2 &= f(t_n + h/2, x_n + hs_1/2) \\ s_3 &= f(t_n + h/2, x_n + hs_2/2) \\ s_4 &= f(t_n + h, x_n + hs_3) \\ x_{n+1} &= x_n + h(s_1 + 2s_2 + 2s_3 + s_4)/6 \\ t_{n+1} &= t_n + h \end{aligned} \tag{1.4}$$

関数文を使っているため、上のアルゴリズムをそのままプログラムに記述することができます。Fig.1.1 に対応する結果を Fig.1.2 と Fig.1.3 に載せますがオイラー法に比べて格段に精度がよくなっている(厳密解に近くなっている)ことがわかります。

program 1-2

```
' 1階常微分方程式 - 2次精度ルンゲ・クッタ法
Sub RK2()
Cells.Clear
```

```

H = 0.1
Tmax = 3#
Nmax = Tmax / H
T = 0#
X = 1#
For N = 0 To Nmax
    Cells(N + 1, 1) = T
    Cells(N + 1, 2) = X
    Cells(N + 1, 3) = Exp(T)
    S1 = F(T, X)
    S2 = F(T + H, X + S1 * H)
    X = X + H * (S1 + S2) / 2#
    T = T + H
Next N
End Sub

Function F(T, X)
    F = X
End Function

```

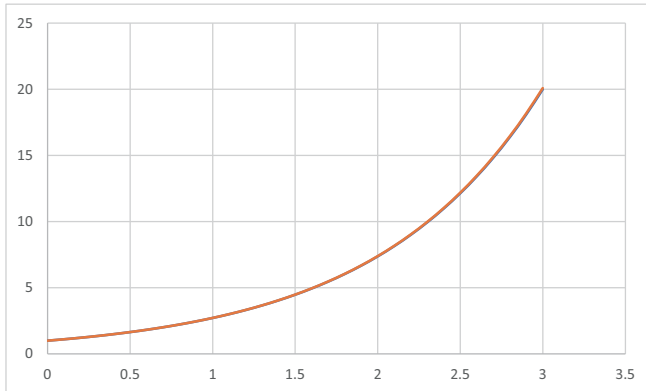


Fig.1.2

program 1-3

```

' 1階常微分方程式 - 4次精度ルンゲ・クッタ法
Sub RK4()
    Cells.Clear
    H = 0.1
    Tmax = 3#
    Nmax = Tmax / H
    T = 0#
    X = 1#
    For N = 0 To Nmax
        Cells(N + 1, 1) = T
        Cells(N + 1, 2) = X
        Cells(N + 1, 3) = Exp(T)
        S1 = F(T, X)
        S2 = F(T + H / 2#, X + S1 * H / 2#)
        S3 = F(T + H / 2#, X + S2 * H / 2#)

```

```

S4 = F(T + H, X + S3 * H)
X = X + H * (S1 + 2# * S2 + 2# * S3 + S4) / 6#
T = T + H
Next N
End Sub

Function F(T, X)
F = X
End Function

```

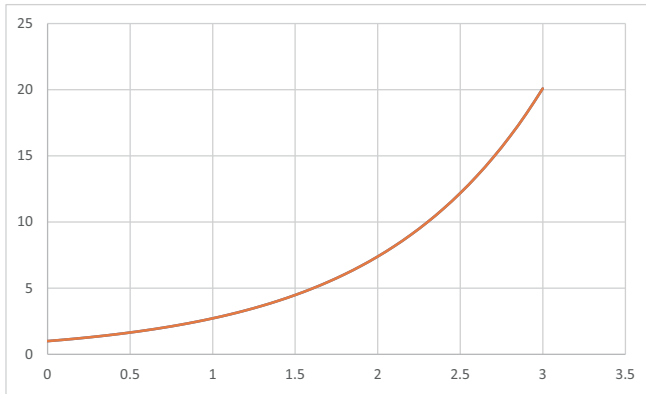


Fig.1.3

1.2 連立微分方程式の初期値問題

正規形の n 階微分方程式は簡単な置きかえによって連立 n 元 1 階微分方程式に書き換えることができます。たとえば 2 階微分方程式の初期値問題

$$\begin{aligned} \frac{d^2x}{dt^2} &= g(t, x, dx/dt) \\ x(0) &= a, \quad x'(0) = b \end{aligned}$$

は $y = dx/dt$ という置き換えによって連立 2 元 1 階微分方程式

$$\begin{aligned} \frac{dx}{dt} &= y \\ \frac{dy}{dt} &= g(t, x, y) \\ x(0) &= a, \quad y(0) = b \end{aligned}$$

に変形できます．これは一般的な連立2元1階微分方程式

$$\frac{dx}{dt} = f(t, x, y), \quad \frac{dy}{dt} = g(t, x, y) \quad (1.5)$$

の特殊な場合になっています．式(1.5)は前節で述べたオイラー法やルンゲ・クッタ法で解くことができます．アルゴリズムはそれぞれ

$$\begin{aligned} s_1 &= f(t_n, x_n, y_n) \\ p_1 &= g(t_n, x_n, y_n) \\ x_{n+1} &= x_n + hs_1 \\ y_{n+1} &= y_n + hp_1 \\ t_{n+1} &= t_n + h \end{aligned} \quad (1.6)$$

$$\begin{aligned} s_1 &= f(t_n, x_n, y_n) \\ p_1 &= g(t_n, x_n, y_n) \\ s_2 &= f(t_n + h, x_n + hs_1, y_n + hp_1) \\ p_2 &= g(t_n + h, x_n + hs_1, y_n + hp_1) \\ x_{n+1} &= x_n + h(s_1 + s_2)/2 \\ y_{n+1} &= y_n + h(p_1 + p_2)/2 \\ t_{n+1} &= t_n + h \end{aligned} \quad (1.7)$$

$$\begin{aligned} s_1 &= f(t_n, x_n, y_n) \\ p_1 &= g(t_n, x_n, y_n) \\ s_2 &= f(t_n + h/2, x_n + hs_1/2, y_n + hp_1/2) \\ p_2 &= g(t_n + h/2, x_n + hs_1/2, y_n + hp_1/2) \\ s_3 &= f(t_n + h/2, x_n + hs_2/2, y_n + hp_2/2) \\ p_3 &= g(t_n + h/2, x_n + hs_2/2, y_n + hp_2/2) \\ s_4 &= f(t_n + h, x_n + hs_3, y_n + hp_3) \\ p_4 &= g(t_n + h, x_n + hs_3, y_n + hp_3) \\ x_{n+1} &= x_n + h(s_1 + 2s_2 + 2s_3 + s_4)/6 \\ y_{n+1} &= y_n + h(p_1 + 2p_2 + 2p_3 + p_4)/6 \\ t_{n+1} &= t_n + h \end{aligned} \quad (1.8)$$