

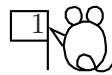
数値計算講義 第8回  
微分方程式の解法 – 初期値問題 –



カーネンコ アレクセイ  
金子 晃

[kanenko@mbk.nifty.com](mailto:kanenko@mbk.nifty.com)  
[alexei.kanenko@docomo.ne.jp](mailto:alexei.kanenko@docomo.ne.jp)  
<http://www.kanenko.com/>

# 常微分方程式の初期値問題



## 1 階常微分方程式の初期値問題

$$\begin{cases} \frac{dx}{dt} = f(t, x), & t_0 \leq t \leq t_{\max}, \\ x(t_0) = c \end{cases}$$

は、区間  $[t_0, t_{\max}]$  を  $N$  等分して  $h = \frac{t_{\max} - t_0}{N}$  とするとき、

前進差分近似  $\frac{x(t+h) - x(t)}{h} = f(t, x(t))$ , すなわち

$$x(t+h) = x(t) + hf(t, x(t))$$

から得られる反復公式

$$x_0 = c,$$

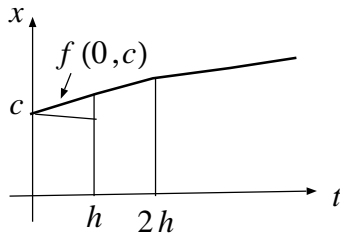
$$x_1 = x_0 + hf(t_0, x_0),$$

.....

$$x_{n+1} = x_n + hf(t_0 + nh, x_n)$$

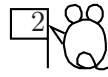
により、容易に解くことができる。

(Euler-Cauchy の折れ線法)



(図は  $t_0 = 0$  の場合)

## 連立常微分方程式の場合



$$\begin{cases} \frac{dx}{dt} = f(t, x, y), & \frac{dy}{dt} = g(t, x, y), & t_0 \leq t \leq t_{\max}, \\ x(t_0) = a, & y(t_0) = b \end{cases}$$

は、次のように離散化する：

$$x_0 = a, \quad y_0 = b$$

$$x_1 = x_0 + hf(t_0, x_0, y_0), \quad y_1 = y_0 + hg(t_0, x_0, y_0)$$

.....

$$x_{n+1} = x_n + hf(t_0 + nh, x_n, y_n), \quad y_{n+1} = y_n + hg(t_0 + nh, x_n, y_n),$$

## 2 階単独常微分方程式の場合

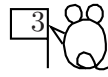
$$\begin{cases} \frac{d^2x}{dt^2} = f(t, x, \frac{dx}{dt}), & t_0 \leq t \leq t_{\max}, \\ x(t_0) = a, & x'(t_0) = b \end{cases}$$

は、そのまま離散化せず、1 階連立常微分方程式

$$\begin{cases} \frac{dx}{dt} = y, & \frac{dy}{dt} = f(t, x, y), & t_0 \leq t \leq t_{\max}, \\ x(t_0) = a, & y(t_0) = b \end{cases}$$

に直してから上の離散化を施すのが普通。

実装例 num8-1.f, num8-2.f



単独方程式の場合，解法のコア部分はたったこれだけ：

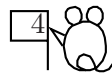
```
H=(TMAX-TMIN)/N      ! 時間の分割数を設定
T=TMIN                ! T を初期時刻に設定
X=C                  ! X を初期値に設定
DO 100 I=1, N
    X=X+F(T,X)*H      ! 次の時刻の X を現在の傾きから計算
    T=T+H              ! 時刻を進める
100 CONTINUE
```

連立方程式の場合もちよっと変わるだけ：

```
H=(TMAX-TMIN)/N      ! 時間の分割数を設定
T=TMIN                ! T を初期時刻に設定
X=A                  ! X を初期値に設定
Y=B                  ! Y を初期値に設定
DO 100 I=1, N
    XS=X+H*F(T,X,Y)   ! 次の時刻の X を計算し保存
    Y=Y+H*G(T,X,Y)   ! 次の時刻の Y を計算
    X=XS              ! 次の時刻の X を保存値から戻す
    T=T+H
100 CONTINUE
```

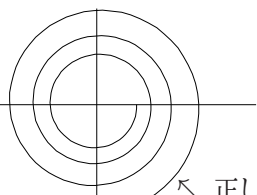
● モデルプログラムは，結果を直接描画するため，xgrf.o をリンクし，DO ループ中の計算値を LINETO 指令で直接描画している．gnuplot で後から描画するときは，DO ループ中で計算した X の値をファイルに書き出し，後で一括描画する．

● 注意 ダミー変数 XS を使うのを忘れないように!

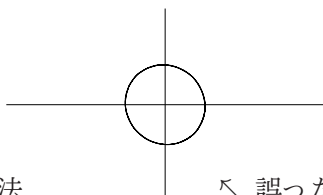


$$\begin{aligned} X &= X + H * F(T, X, Y) \\ Y &= Y + H * G(T, X, Y) \end{aligned}$$

と書くと, Euler-Cauchy の折れ線法の正しい実装ではなくなる.  
間違えるとどうなるか単振動の連立化方程式で観察してみよう (num8-2a.f).  
 $h = 0.1, x(0) = 1.0, y(0) = 0.0$  の場合, 200 ステップ分を描画すると

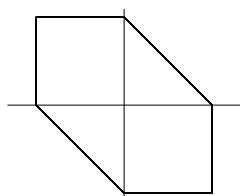
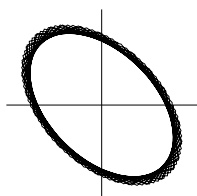
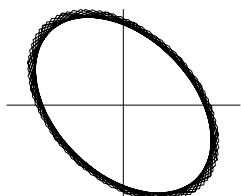


正しい Euler-Cauchy 法



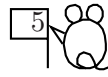
誤った方法

どうも話がうますぎると, 後者の軌道を  $x(0) = 3, y(0) = 0$  に拡大して,  
 $h = 0.8, h = 0.9, h = 1.0$  でそれぞれ 100 ステップやってみると



一周当たりの誤差のサイズは正しい実装と同じ程度だが,  
軌道を発散させないような不思議な保存性を持つ!

## Euler-Cauchy 法の誤差評価



$|f(t, x)| \leq M_0$  及び, **Lipschitz 条件**

$$|f(t, x) - f(s, y)| \leq m_1|t - s| + M_1|x - y|$$

を仮定し, 1 ステップ当たりの近似解と真の解との差の拡大を見る.

真の解  $x(t)$  は, 積分方程式

$$x(t_{n+1}) = x(t_n) + \int_{t_n}^{t_{n+1}} f(t, x(t)) dt$$

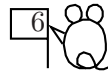
を満たしている.

他方, 近似解は,

$$x_{n+1} = x_n + hf(t_n, x_n) = x_n + \int_{t_n}^{t_{n+1}} f(t_n, x_n) dt$$

を満たしている. よって差は,

$$\begin{aligned} & |x(t_{n+1}) - x_{n+1}| \\ & \leq |x(t_n) - x_n| + \left| \int_{t_n}^{t_{n+1}} f(t, x(t)) dt - \int_{t_n}^{t_{n+1}} f(t_n, x_n) dt \right| \\ & \leq |x(t_n) - x_n| + \int_{t_n}^{t_{n+1}} |f(t, x(t)) - f(t_n, x_n)| dt \end{aligned}$$



ここで Lipschitz 条件等を用いて、積分記号下を

$$\begin{aligned} |f(t, x(t)) - f(t_n, x_n)| &\leq m_1(t - t_n) + M_1|x(t) - x_n| \\ &\leq m_1(t - t_n) + M_1|x(t_n) - x_n| + M_1|x(t) - x(t_n)| \end{aligned}$$

と分解すると、最後の項に

$$|x(t) - x(t_n)| = \left| \int_{t_n}^t f(s, x(s)) ds \right| \leq M_0(t - t_n)$$

(ここに  $M_0$  は考えている領域での  $f$  の上限) を用いて、結局

$$|f(t, x(t)) - f(t_n, x_n)| \leq (m_1 + M_0M_1)(t - t_n) + M_1|x(t_n) - x_n|$$

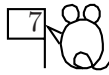
よって、最終的に

$$\begin{aligned} |x(t_{n+1}) - x_{n+1}| \\ \leq |x(t_n) - x_n| + \frac{m_1 + M_0M_1}{2}(t_{n+1} - t_n)^2 + M_1(t_{n+1} - t_n)|x(t_n) - x_n| \end{aligned}$$

i.e.

$$|x(t_{n+1}) - x_{n+1}| \leq (1 + M_1h)|x(t_n) - x_n| + \frac{m_1 + M_0M_1}{2}h^2$$

という 1 段分に対する誤差の伝播公式が得られた。



これを  $n = 1, \dots, N - 1$  について反復適用する. あるいは,

$$|x(t_N) - x_N| \leq (1 + M_1 h) |x(t_{N-1}) - x_{N-1}| + \frac{m_1 + M_0 M_1}{2} h^2$$

$$(1 + M_1 h) |x(t_{N-1}) - x_{N-1}| \leq (1 + M_1 h)^2 |x(t_{N-2}) - x_{N-2}| + (1 + M_1 h) \frac{m_1 + M_0 M_1}{2} h^2$$

.....

$$(1 + M_1 h)^{N-1} |x(t_1) - x_1| \leq (1 + M_1 h)^N |x(t_0) - x_0| + (1 + M_1 h)^{N-1} \frac{m_1 + M_0 M_1}{2} h^2$$

を総和して, 最後の項に等比級数の和の公式を使うと

$$|x(t_N) - x_N| \leq (1 + M_1 h)^N |x(t_0) - x_0| + \frac{(1 + M_1 h)^N - 1}{M_1 h} \frac{m_1 + M_0 M_1}{2} h^2$$

ここで  $Nh = t_{\max} - t_0 =: T$  は  $h$  によらぬ定数,  
 また  $(1 + M_1 h)^N = \{(1 + M_1 h)^{1/M_1 h}\}^{M_1 T} \leq e^{M_1 T}$  より,

$$|x(t_N) - x_N| \leq e^{M_1 T} |x(t_0) - x_0| + \frac{m_1 + M_0 M_1}{2 M_1} e^{M_1 T} h$$

が得られ, 1 次の近似であることが確定した.

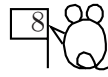
普通は, 初期値には誤差は無いものとし, 右辺の第1 項を省く.  
 初期値に誤差が有るときは, 各段の計算にも丸め誤差を入れるのが妥当で,  
 この場合は上の評価は

$$|x(t_N) - x_N| \leq e^{M_1 T} |x(t_0) - x_0| + \frac{e^{M_1 T}}{M_1} \left( \frac{m_1 + M_0 M_1}{2} h + \frac{\varepsilon}{h} \right)$$

となる.



## 常微分方程式に対する種々の問題



### ● 初期値問題

$$\begin{cases} \frac{d^2u}{dx^2} + q(x)u = f(x), \\ u(0) = u_0, \quad \frac{du}{dx}(0) = u_1 \end{cases}$$

離散化はもっとも簡単で、逐次代入で解ける。

### ● 境界値問題

$$\begin{cases} \frac{d^2u}{dx^2} + q(x)u = f(x), \\ u(0) = 0, \quad u(1) = 0 \end{cases}$$

離散化は連立一次方程式を解くこと。  $A\mathbf{u} = \mathbf{f}$

### ● 固有値問題

$$\begin{cases} \frac{d^2u}{dx^2} + q(x)u = \lambda u, \\ u(0) = 0, \quad u(1) = 0 \end{cases}$$

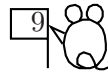
離散化は行列の固有値問題となる。  $A\mathbf{u} = \lambda\mathbf{u}$

Cf.  $A$  が有限行列のとき、

$A\mathbf{u} = \lambda\mathbf{u} + \mathbf{f}$  が一意に解ける  $\iff \lambda$  が行列  $A$  の固有値でない。

境界条件付きの線型常微分作用素は無次元の行列だが、同じことが成り立つ。

## 2 次の Runge-Kutta 法 num8-3.f



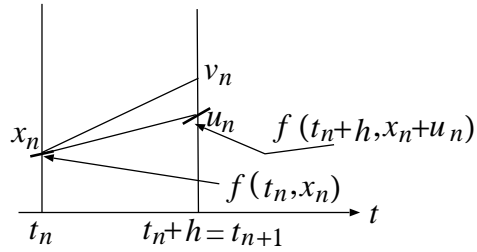
Euler 法は、右辺が未知函数  $x$  を含まないとき、すなわち単なる定積分のときは、Riemann 近似和に対応し、はなはだ近似度が悪い。これを台形公式に相当するものに改良したのが、2 次の Runge-Kutta 法:

$$\begin{aligned}u_n &= hf(t_n, x_n), \\v_n &= hf(t_{n+1}, x_n + u_n), \\x_{n+1} &= x_n + \frac{1}{2}(u_n + v_n)\end{aligned}$$

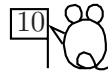
時刻  $t_n$  における傾きで予測した到達点での傾きで進み直したものと平均すると覚えておくとよい。

実装の核心部分は

$$\begin{aligned}U &= H * F(T, X) \\V &= H * F(T+H, X+U) \\X &= X + (U+V) / 2\end{aligned}$$



#### 4 次 Runge-Kutta 法 num8-4.f



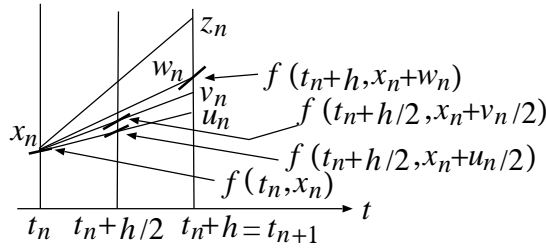
これは定積分の Simpson 公式に相当するもので、  
普通 Runge-Kutta 法と呼ばばこれを指す。

$x_n$  から  $x_{n+1}$  の更新アルゴリズムは次の通り：

$$\begin{aligned} u_n &= hf(t_n, x_n), \\ v_n &= hf(t_n + h/2, x_n + u_n/2), \\ w_n &= hf(t_n + h/2, x_n + v_n/2), \\ z_n &= hf(t_n + h, x_n + w_n), \\ x_{n+1} &= x_n + u_n/6 + v_n/3 + w_n/3 + z_n/6 \end{aligned}$$

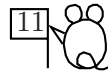
実装するときは、 $u_n, v_n, w_n, z_n$  は共通の臨時変数を使えばよいので、

```
U=H*F(T,X)
XS=X+U/6
U=H*F(T+H/2,X+U/2)
XS=XS+U/3
U=H*F(T+H/2,X+U/2)
XS=XS+U/3
U=H*F(T+H,X+U)
X=XS+U/6
```



※ H2=H/2 が予め代入されている (少しでも計算を速くするけちな手法)

## Runge-Kutta 法の近似のオーダー



Runge-Kutta 法の打ち切り誤差の評価は非常に面倒で、書かれている書物はほとんど無い。

よく書かれているのはオーダーだけの評価だが、それも4次の公式は大変。ここでは2次の公式に対して、オーダー評価を示す。

$$u_n = hf(t_n, x_n),$$

$$\begin{aligned} v_n &= hf(t_n + h, x_n + u_n) \\ &= hf(t_n, x_n) + h^2 f_t(t_n, x_n) + hu_n f_x(t_n, x_n) + O(h^3) \\ &= hf(t_n, x_n) + h^2 f_t(t_n, x_n) + h^2 f(t_n, x_n) f_x(t_n, x_n) + O(h^3) \end{aligned}$$

$$\begin{aligned} \therefore x_{n+1} &= x_n + \frac{1}{2}(u_n + v_n) \\ &= x_n + hf(t_n, x_n) + \frac{h^2}{2} f_t(t_n, x_n) + \frac{h^2}{2} f(t_n, x_n) f_x(t_n, x_n) + O(h^3). \end{aligned}$$

他方、 $x(t)$  を真の解として

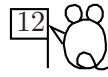
$$\begin{aligned} f(s, x(s)) &= f(t_n, x(t_n)) + (s - t_n) f_t(t_n, x(t_n)) \\ &\quad + (x(s) - x(t_n)) f_x(t_n, x(t_n)) + O((s - t_n)^2) \\ &= f(t_n, x(t_n)) + (s - t_n) f_t(t_n, x(t_n)) \\ &\quad + (s - t_n) f(t_n, x(t_n)) f_x(t_n, x(t_n)) + O((s - t_n)^2) \end{aligned}$$

を積分して、

$$\begin{aligned} x(t_{n+1}) &= x(t_n) + hf(t_n, x(t_n)) + \frac{h^2}{2} f_t(t_n, x(t_n)) \\ &\quad + \frac{h^2}{2} f(t_n, x(t_n)) f_x(t_n, x(t_n)) + O(h^3) \end{aligned}$$

よって両者の差をとり、 $f, f_t, f_x$  の中の  $x(t_n)$  を  $x_n$  で置き換えると

$$|x_{n+1} - x(t_{n+1})| \leq |x_n - x(t_n)| + Mh|x_n - x(t_n)| + O(h^3)$$



$$\leq (1 + Mh)|x_n - x(t_n)| + O(h^3)$$

これを  $n = N, N-1, \dots, 1$  について書き並べ

$$|x_N - x(t_N)| \leq (1 + Mh)|x_{N-1} - x(t_{N-1})| + O(h^3)$$

$$(1 + Mh)|x_{N-1} - x(t_{N-1})| \leq (1 + Mh)^2|x_{N-2} - x(t_{N-2})| + (1 + Mh)O(h^3)$$

.....

$$(1 + Mh)^{N-1}|x_1 - x(t_1)| \leq (1 + Mh)^N|x_0 - x(t_0)| + (1 + Mh)^{N-1}O(h^3)$$

辺々加えると,  $N = (b-a)/h$ ,  $(1 + Mh)^{1/Mh} \leq e$  に注意して

$$|x_N - x(t_N)| \leq (1 + Mh)^N|x_0 - x(t_0)| + \frac{(1 + Mh)^N - 1}{Mh}O(h^3)$$

$$\leq e^{M(b-a)}|x_0 - x(t_0)| + \frac{e^{M(b-a)} - 1}{M}O(h^2)$$

を得る.

※ 以上の評価では  $f_x(t, x)$  の  $x$  に関する Lipschitz 連続性を使うので,

例えば  $f_{xx}(t, x)$  が有界なら OK.

上で  $O(h^3)$  と書いた部分は, この定数を使えば, 見積り可能なある

定数により  $Kh^3$  で抑えられ, 最後の  $O(h^2)$  も  $Kh^2$  にできる.

4 次の Runge-Kutta 公式に対しても同様にして 1 段の誤差が  $O(h^5)$ ,  
従って全体としての誤差が  $O(h^4)$  であることが理論的に示せる.

(教科書の第 8 章参照.)

## 平面の力学系

1 階連立常微分方程式は、力学系 (dynamical system) とも呼ばれる。<sup>13</sup>  
粒子が方程式で記述される法則に従い、時間とともに位置を変える様子 (流れ) を研究する学問である。

この言葉の起源は Newton の運動方程式:

直線上を保存力の場  $f(x)$  に従って運動する質点は  $m \frac{d^2x}{dt^2} = f(x)$  という

方程式を満たす。運動量座標  $p = m \frac{dx}{dt}$  を導入すると、1 階連立系:

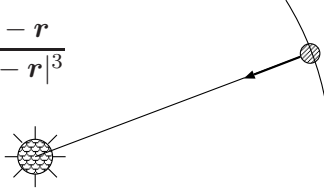
$$\begin{cases} \frac{dx}{dt} = \frac{p}{m}, \\ \frac{dp}{dt} = f(x) \end{cases}$$

これは相空間 (phase space) と呼ばれる  $xp$  空間の流れを記述する。

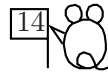
実用的な力学系の例: 惑星の運動 (2 体問題)。

$$M \frac{d^2 \mathbf{R}}{dt^2} = \gamma M m \frac{\mathbf{r} - \mathbf{R}}{|\mathbf{R} - \mathbf{r}|^3}, \quad m \frac{d^2 \mathbf{r}}{dt^2} = \gamma M m \frac{\mathbf{R} - \mathbf{r}}{|\mathbf{R} - \mathbf{r}|^3}$$

前者が太陽, 後者が惑星とすれば  $M \gg m$ .



$$M \frac{d^2 \mathbf{R}}{dt^2} = \gamma M m \frac{\mathbf{r} - \mathbf{R}}{|\mathbf{R} - \mathbf{r}|^3}, \quad m \frac{d^2 \mathbf{r}}{dt^2} = \gamma M m \frac{\mathbf{R} - \mathbf{r}}{|\mathbf{R} - \mathbf{r}|^3}$$



二つの方程式を加えると

$$\frac{d^2}{dt^2}(M\mathbf{R} + m\mathbf{r}) = 0 \quad \therefore \frac{d}{dt}(M\mathbf{R} + m\mathbf{r}) = \text{一定}$$

(重心  $\frac{M\mathbf{R} + m\mathbf{r}}{M + m}$  の等速度運動を表現する式.)

重心座標を導入し、重心からの相対運動を考察する：

$$\mathbf{r}_G = \mathbf{r} - \frac{M\mathbf{R} + m\mathbf{r}}{M + m} = \frac{M}{M + m}(\mathbf{r} - \mathbf{R})$$

と置けば

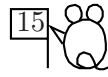
$$m \frac{d^2 \mathbf{r}_G}{dt^2} = m \frac{d^2 \mathbf{r}}{dt^2} = \gamma M m \frac{\mathbf{R} - \mathbf{r}}{|\mathbf{R} - \mathbf{r}|^3} = -\gamma \frac{M^3 m}{(M + m)^2} \frac{\mathbf{r}_G}{|\mathbf{r}_G|^3}$$

これは、相空間が4次元内の流れだが、空間座標だけをリアルタイムで描画すると2Dで表現できる (`planet.f`).

(更に、解析的手段で本当の平面の力学系に帰着できる.)

この連立常微分方程式を手計算で解いて実際に楕円軌道を得ると、Newtonの感激を追体験できるので、ぜひやってみよう!

**Van der Pol 方程式** 平面の力学系の実用的な代表例として  
緩和振動を記述する有名な方程式. (vanderpol.f)

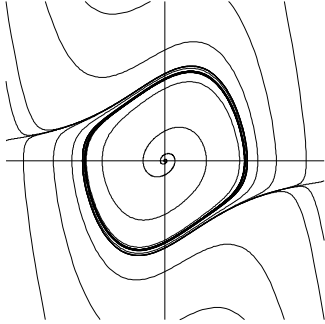


$$x'' - \varepsilon(1 - x^2)x' + x = 0,$$

一階連立系としては

$$\begin{cases} x' = y, \\ y' = -x + \varepsilon(1 - x^2)y \end{cases}$$

$|x| > 1$  のときはブレーキをかけ,  $|x| < 1$  のときはアクセルをかけることにより,  
ちょうど周期  $2\pi$  の安定な軌道を維持させる仕組み.





## 高次元の力学系

物理法則は一般に時間に依存しないので、力学系の右辺は  $t$  を含まないのが普通。

このような方程式系は自励系と呼ばれる。

自励系では、解軌道は時刻の平行移動で不変。

従って初期値問題の解の一意性により、二つの解軌道は交わらない。

特に、平面の力学系は軌道が互いに他を分離するので、構造が簡単となる。

これに対し、3次元以上になると、一方の軌道が他方に巻き付いたりでき、非常に複雑な軌道が現れる。

高次元力学系で古来有名なのが3体問題で、未だに分からないことが多い。つい数年前にも新たな周期解 (八の字解) が発見されたりしている。

また、偏微分方程式の離散化で乱流のモデル等として得られた方程式系の中には、カオス的な動きをするものがあり、決定論的カオスと呼ばれる。

(Newton 力学では、初期位置と初期速度を指定すれば、質点のその後の運動は一意に決まってしまう (決定論的)。

それなのに、その振舞が複雑で、予測不可能に見える (カオス的) ので、こう呼ばれる。)

プログラム例 `lorenz.f`, `rossler.f` をコンパイルし実行して決定論的カオスの世界を味わってみよ。

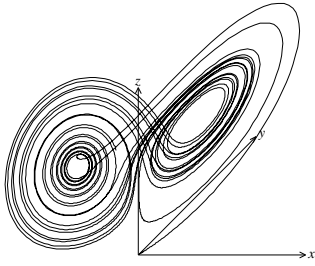
**Lorenz の方程式**

気象学者が考え出した、気体の運動方程式の簡略化モデルで、

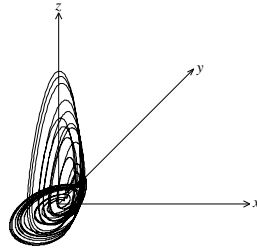
$$x' = \sigma(y - x), \quad y' = x(R - z) - y, \quad z' = xy - bz$$

**Rössler の方程式**は、それを更に単純化した

$$x' = -y - z, \quad y' = x + ay, \quad z' = bx + z(x - c)$$



Lorenz アトラクタ



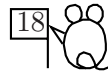
Rössler アトラクタ

どちらも単なる Euler-Cauchy 法で書かれている。

これらはカオス的な現象が見えて来るまで軌道を追跡すると、かなりの誤差が累積するはずだが、もっともらしい軌道に見えるのはなぜか？

どの軌道もアナログ的には似たような振舞をするので、軌道は変わっても人間には感知できない。

あるいみではインチキを見せている！



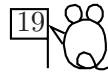
- 1 初期値問題  $\begin{cases} \frac{dx}{dt} = x, \\ x(0) = 1 \end{cases}$  を解析的に解け. またこの近似解を

求める Euler 法のプログラム num8-1.f をコンパイル・実行し, 分割数をいろいろ変えて近似のオーダーを数値的に観察せよ. できれば丸め誤差の影響も考慮し, 最良の刻み幅  $h$  を推測せよ.

- 2 2次元力学系の num8-2.f および num8-2a.f をコンパイル・実行し, 種々の  $h$  に対して挙動の違いを観察せよ.

- 3 Runge-Kutta 法のプログラム num8-3.f, num8-4.f をコンパイル・実行し, 分割数をいろいろ変えて近似のオーダーを調べよ. できれば丸め誤差の影響も考慮し, 最良の刻み幅  $h$  を推測せよ.

- 4 planet.f, vanderpol.f, lorenz.f, rossler.f, eight.c をコンパイル・実行し, 出力結果を味わえ. sphereplanet.c, sphereeight.c はそれぞれ planet.f eight.c の OpenGL 版です. OpenGL は伊藤先生の『コンピュータグラフィックス』で後期に学べますが, 一足早く opengl-gcc sphereeight.c 等でコンパイルして味わってみましょう. (opengl-gcc はシェルスクリプトです. 伊藤先生は Makefile にしています.)



問題 8.1 単振動の方程式を1階連立化した

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

に対し、エネルギー保存則  $x^2 + y^2 = \text{const.}$  を示せ.

(解の具体的な函数形を用いず、微分方程式から導け.)

課題 8.2 単振動の方程式を1階連立化した上記の方程式に、ダミー変数を

使い忘れた誤った実装をした場合には、かなり大きい  $h$  についても、

軌道が無限に発散しないという現象を数学的に正当化せよ.

また正しい Euler-Cauchy 法では、この量は時間ステップとともに

どう変化するか述べよ. [ ヒント: 第  $n$  ステップの近似解  $(x_n, y_n)$  に対し、

量  $x_n^2 + hx_n y_n + y_n^2$  が  $n$  によらず一定となることを計算で示せ. ]

課題 8.3 振幅の大きな振り子の運動方程式は、 $\frac{d^2x}{dt^2} = -k \sin x$  となり、

初等関数では解くことができない. この近似解を求める方法を述べよ.

(初期条件としては、振り子を振れ角  $30^\circ$  の位置で静かに離れた場合

を仮定せよ. 実際に動くプログラムまで書くには及ばない.)

問題 8.4 常微分方程式の境界値問題

$$-\frac{d^2u}{dx^2} + q(x)u = f(x) \quad (0 < x < 1), \quad u(0) = u(1) = 0$$

を離散化して解くとき、どのような計算が必要となるか? ただし、

$q, f$  は既知の関数とする.

問題 8.5 Runge-Kutta 法について述べよ. また、これを最も簡単な微分

方程式  $y' = f(x)$  に適用すると、 $f(x)$  に対するどんな積分公式となるか?

問題 8.6 微分方程式  $\frac{dy}{dx} = y$  の初期条件  $y(0) = 1$  を満たす解の  $x = 1$

における近似値を、メッシュ幅  $h = 1.0$  について Runge-Kutta 法で

計算した値 (すなわち、Runge-Kutta 法の1段だけの計算値) を示せ.