

2020(令和2)年度 N体シミュレーション雨水の学校

常微分方程式の数値解法 の基礎の初歩の触りの一部

令和3年2月16日

伊藤孝士 (国立天文台天文シミュレーションプロジェクト)

話の内容

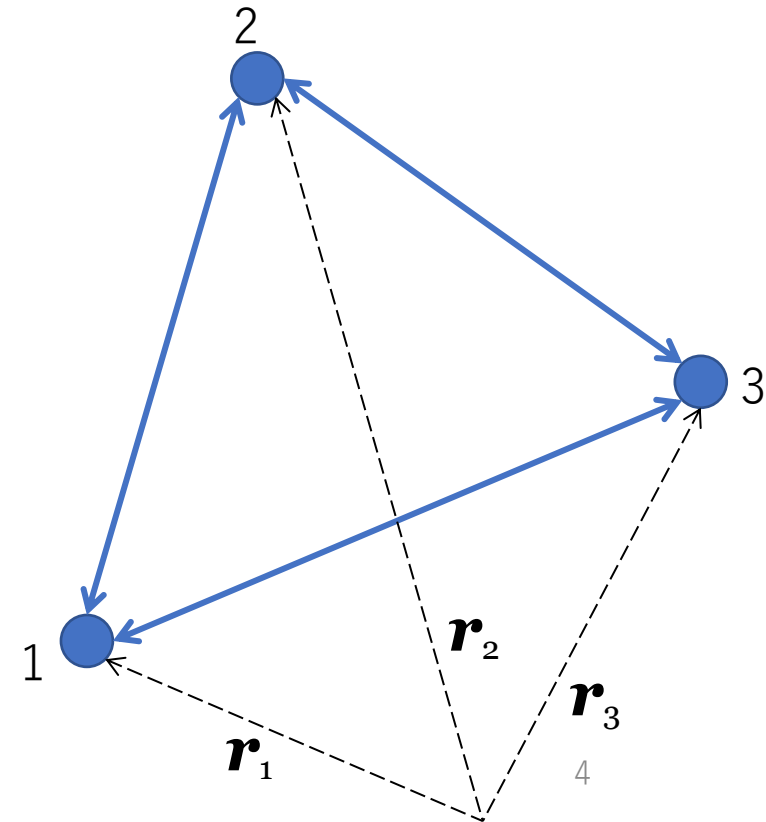
- 一般論と前置き
- 汎用的な解法群
- 正準形式の解法

一般論と前置き

(重力)N体問題の運動方程式

$$m_i \frac{d^2 \mathbf{r}_i}{dt^2} = -G \sum_{\substack{j=1 \\ j \neq i}}^N m_i m_j \frac{\mathbf{r}_i - \mathbf{r}_j}{|\mathbf{r}_i - \mathbf{r}_j|^3}$$

- この講義で話すのは時間積分の部分
- 右辺(力)の計算については別講義で



常微分方程式の正規形

または

$$\frac{dy}{dx} = f(x, y)$$
$$y' = f(x, y)$$

x が時刻 t なら時間発展

$$\frac{dy}{dt} = f(t, y)$$

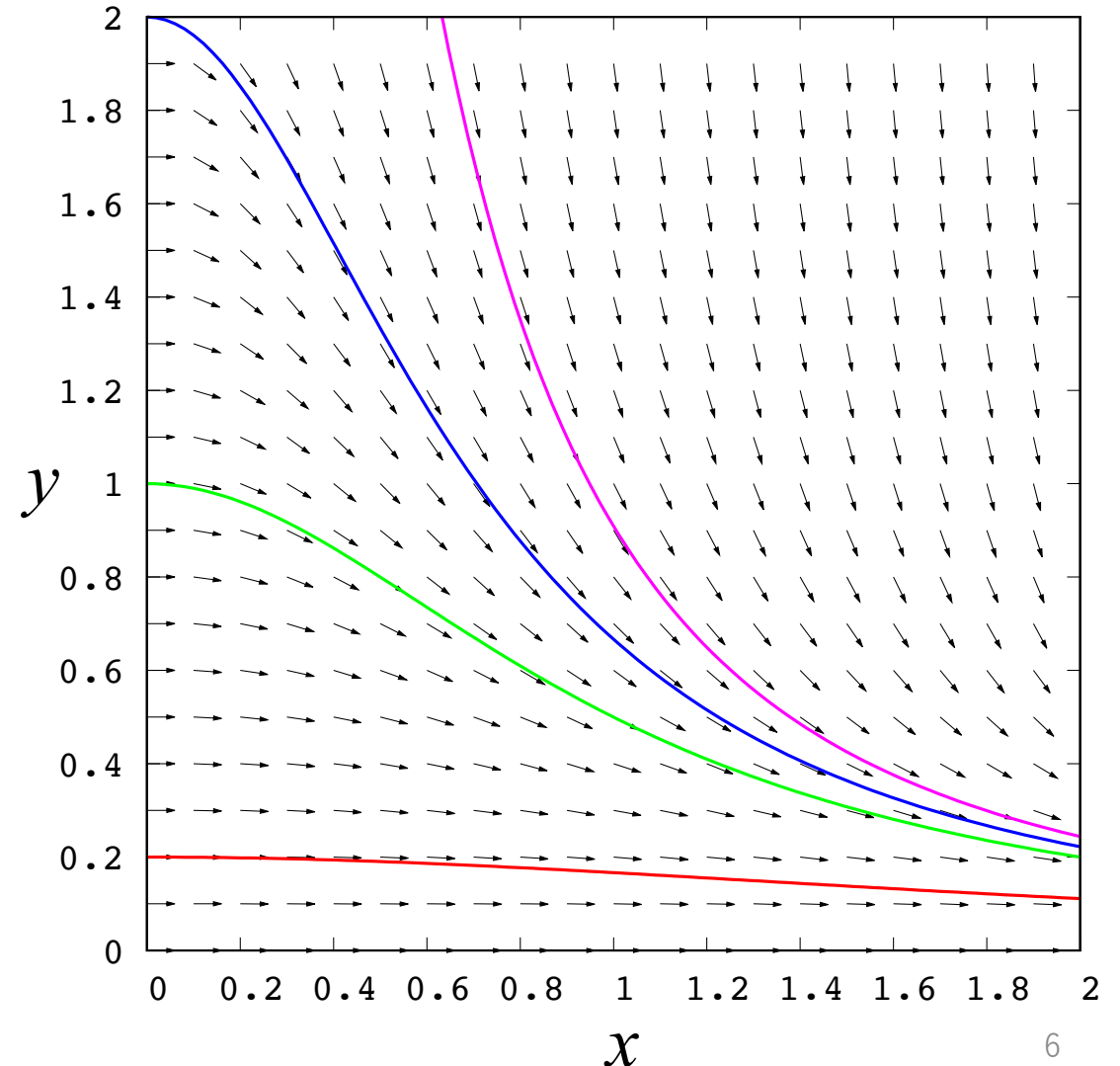
この業界では以下の形で示されることあり

$$\frac{dx}{dt} = f(t, x)$$

が、これは市販の多くの教科書と整合しない。
今回はあくまで y を従属変数とする。

微分方程式の初期値問題

- $y' = f(x, y)$
- 「方向の場」
 - 初期値を与えればあとはそこからの流れに乗って進む
- 例. $y' = -2xy^2$
 - 厳密解 $y = 1/(x^2 + C)$
 - 各色線は異なる初期値から



階数の下げ方

- 高階の微係数 $y, y', y'', y''', \dots, y^{(n)}$ を含む常微分方程式
 - $y^{(n)} = f(x; y, y', y'', y''', \dots, y^{(n-1)})$
- 0階, 1階, \dots , $n-1$ 階の微係数を別個の従属変数と見做す
 - $y \rightarrow y_0, y' \rightarrow y_1, y'' \rightarrow y_2, \dots, y^{(n-2)} \rightarrow y_{n-2}, y^{(n-1)} \rightarrow y_{n-1}$
- 新しい従属変数で書き直した微分方程式
 - $y'_{n-1} = f(x; y_0, y_1, y_2, \dots, y_{n-1})$ ← y_{n-1} に関する一階の常微分方程式
 - $y'_{n-2} = y_{n-1}$ ← y_{n-2} に関する一階の常微分方程式
 - \vdots
 - $y'_1 = y_2$
 - $y'_0 = y_1$
 - 一本の n 階の常微分方程式 → n 本の連立一階常微分方程式
- 「階数」 = order

階数の下げ方

例. 三階の常微分方程式

$$y''' + 3yy'' = 0$$

変数の置き換え

$$y \rightarrow y_0, \quad y' \rightarrow y_1, \quad y'' \rightarrow y_2$$

以下の連立一階常微分方程式に変換し、解く

$$\begin{cases} y_2' = -3y_0y_2, \\ y_1' = y_2, \\ y_0' = y_1 \end{cases}$$

$$m_i \frac{d^2 \mathbf{r}_i}{dt^2} = -G \sum_{\substack{j=1 \\ j \neq i}}^N m_i m_j \frac{\mathbf{r}_i - \mathbf{r}_j}{|\mathbf{r}_i - \mathbf{r}_j|^3}$$

重力N体問題では

運動方程式は変数 \mathbf{r} に関する二階の常微分方程式

$$\frac{d^2 \mathbf{r}}{dt^2} = f(t; \mathbf{r})$$

速度 \mathbf{v} を新たな従属変数とし、階数を下げる

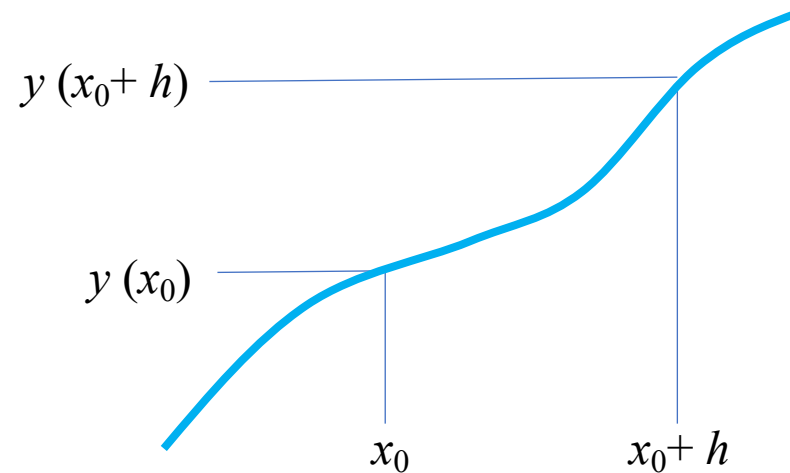
$$\left\{ \begin{array}{l} \frac{d\mathbf{v}}{dt} = f(t; \mathbf{r}), \\ \frac{d\mathbf{r}}{dt} = \mathbf{v} \end{array} \right.$$

常微分方程式を「解く」

$y' = f(x, y)$ を「解く」とは？

- 形式的には以下を求め続けること

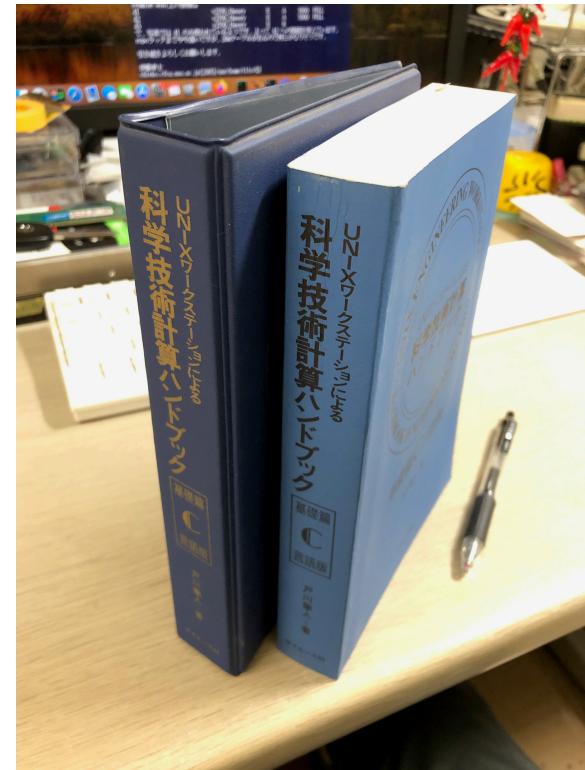
$$y(x_0 + h) = y(x_0) + \int_{x_0}^{x_0+h} f(x, y(x)) dx$$



汎用的な解法群

汎用的な解法群

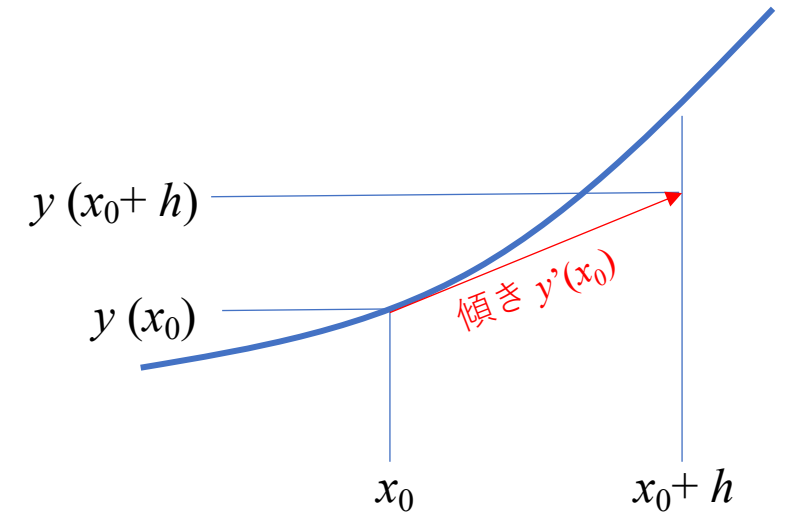
- 長い研究史があり、多くの成果があり、とても奥深い
 - Euler法
 - Runge-Kutta法
 - 多段法、予測子・修正子法
 - 陽解法と陰解法
 - :
- この講義(前半部)が沿う書籍
- N体学校の過去の講義録も参照



簡単な方法

その場の傾きから次の値を予想する

$$y(x_0 + h) = y(x_0) + hf(x_0, y(x_0))$$



テイラー展開の一次項と等価

$$y(x_0 + h) = y(x_0) + \frac{h}{1!}y'(x_0) + \frac{h^2}{2!}y''(x_0) + \frac{h^3}{3!}y'''(x_0) + \dots$$

「オイラー法」

オイラー法の精度

1ステップ(h)あたりの誤差

- $O(h^2)$

$x = a$ から $x = b$ をステップ h で進む

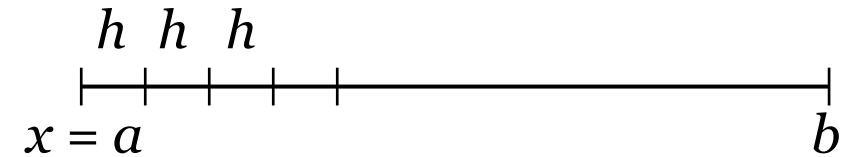
- ステップの総数 = $(b-a) / h$

全ステップで発生する誤差

- $O(h^2) * (b-a) / h \rightarrow O(h^1)$

→ オイラー法の精度は一次 $O(h^1)$

- 「次数」 = order (時に degree)



実例

- 常微分方程式 $y' = y$
- 初期条件 $(x, y) = (0, 1)$
- 厳密解あり $y = e^x$

```
BEGIN{
  #dx   = 0.10  # from the command line
  x0    = 0.0   # initial value
  y0    = 1.0   # initial value
  xmax  = 10.0
  printf("%e %e %e\n", x0, y0, 0)
}

{
  x = x0
  y = y0
  while (x < xmax){
    y += fxy(x,y)*dx
    x += dx
    printf("%e %e %e\n", x, y, fabs(y-yas(x)))
  }
}
```

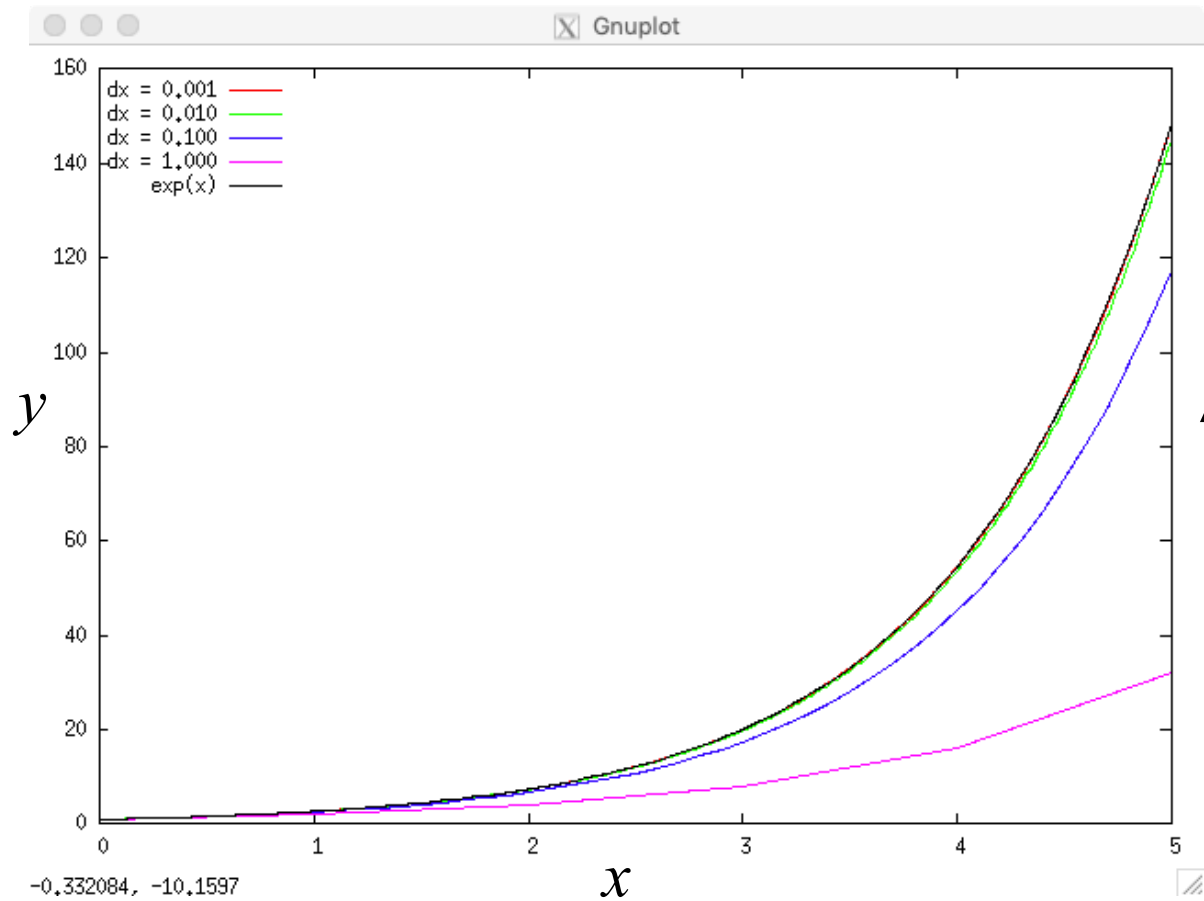
```
# y' = dy/dx = f(x,y)
function fxy(x,y){
  return y
}

# analytic solution of y(x)
function yas(x){
  return exp(x)
}

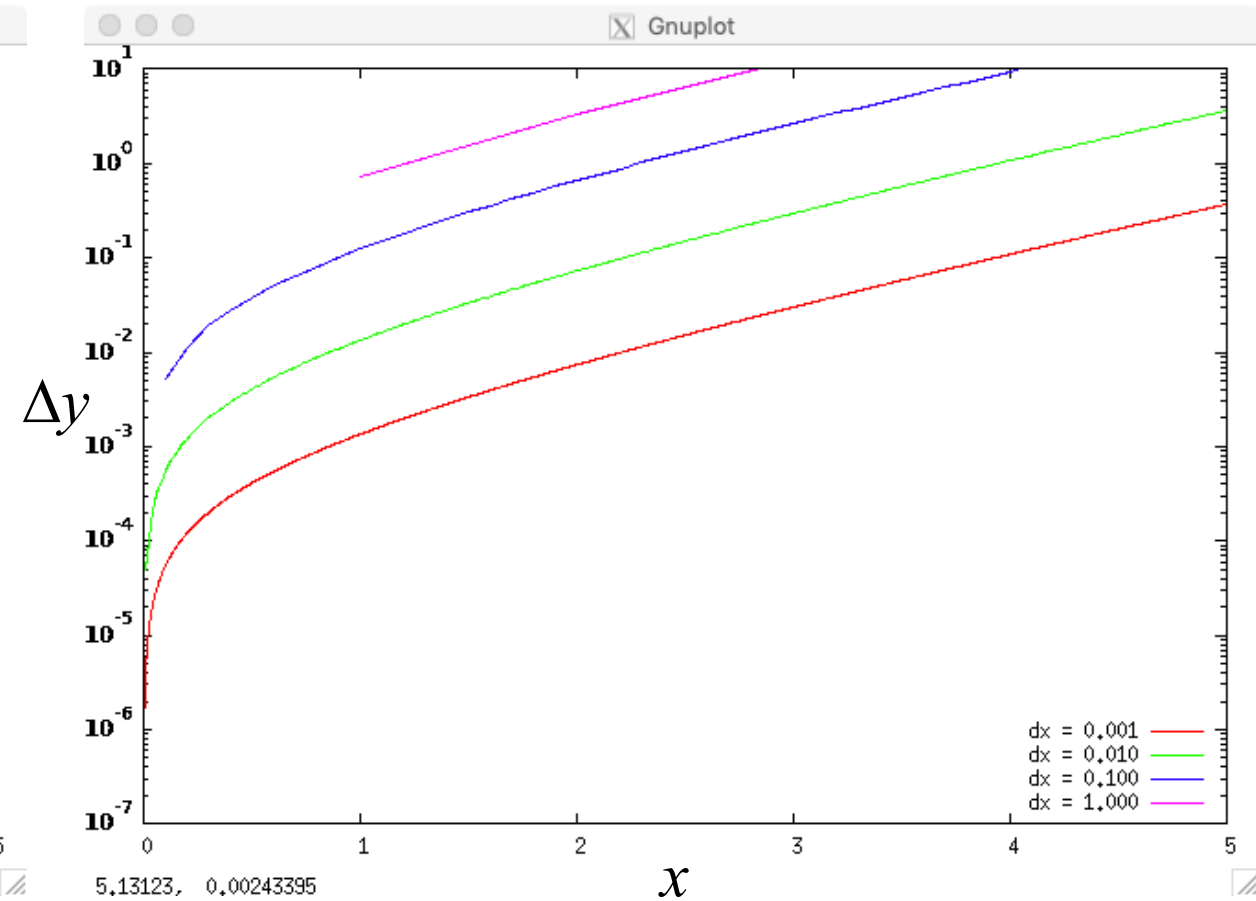
# absolute value
function fabs(x){
  if (x >= 0) return x
  else      return -x
}
```

実例

$y = e^x$ との比較



$y = e^x$ との差分

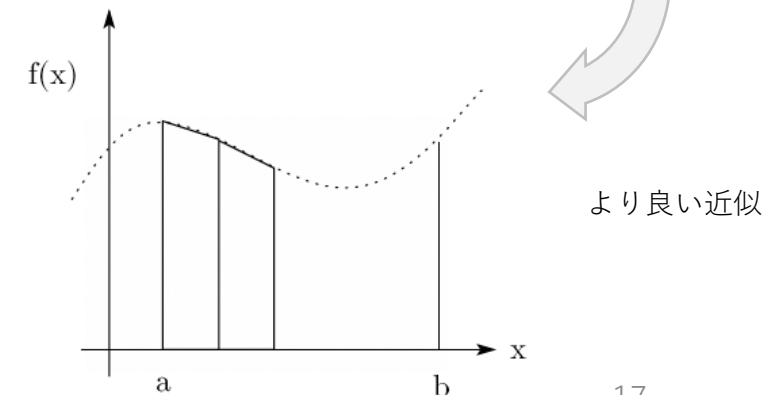
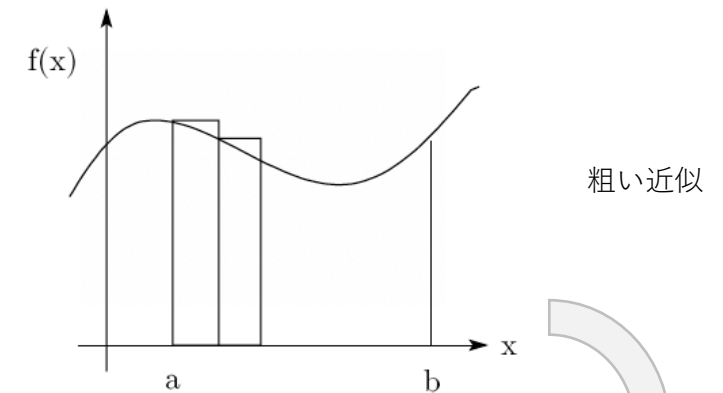


(一般論) 数値計算の誤差

- 物理法則の表現は連続関数
- だが計算機内での操作は離散的
 - 計算の打ち切り = 妥協
 - 無限級数をどこかの次数で打ち切る
 - 曲線的な図形を四角形で近似する
 - :
- この操作による誤差 → 打ち切り誤差
 - 本講義が言及する誤差(精度)は専らこれ

$$y(x_0 + h) = y(x_0) + \frac{h}{1!}y'(x_0) + \frac{h^2}{2!}y''(x_0) + \frac{h^3}{3!}y'''(x_0) + \dots$$

無限級数を
どこかで打ち切る



(一般論) 数値計算の誤差

計算機内の小数点表現は有限桁

- $4.6875 = 4 + 0.5 + 0.125 + 0.0625$
 $= 1*2^2 + 0*2^1 + 0*2^0 + 1*2^{-1} + 0*2^{-2} + 1*2^{-3} + 1*2^{-4}$
→ 100.1011 (二進法表現)

有限のビット数では有限個の数値しか表現できない

- 1ビット → $2^1 = 2$ 種類
- 2ビット → $2^2 = 4$ 種類
- :
- 7ビット → $2^7 = 128$ 種類
- :

(一般論) 数値計算の誤差

現代の標準規格 [IEEE754](#)

- 単精度 (24ビット相当)
 - $2^{24} = 16777216$ 種類 $\rightarrow \log_{10}(2^{24}) \sim 7.22$ 桁
- 倍精度 (53ビット相当)
 - $2^{53} = 9007199254740992$ 種類 $\rightarrow \log_{10}(2^{53}) \sim 15.95$ 桁

本物の実数は無限にある

- 計算機ではそのうち幾つかしか表現できない
- 近い数値で代表させる \rightarrow 「丸める」 \rightarrow [丸め誤差](#)

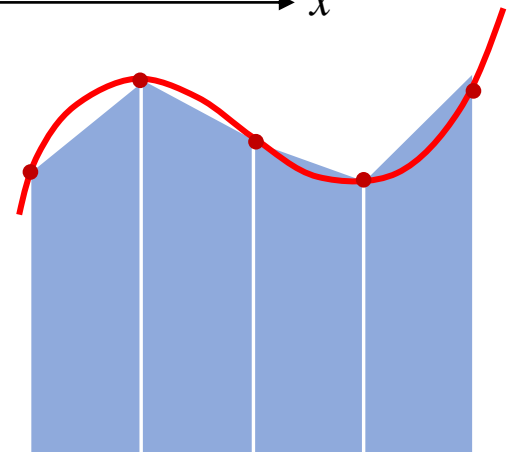
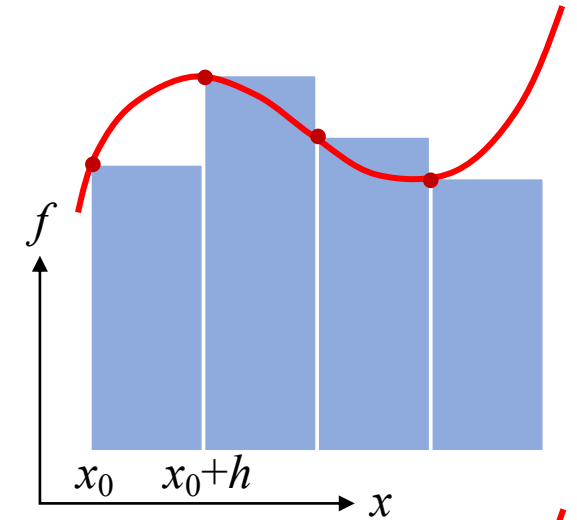
(オイラー法) 精度の向上

- 形式的な解
$$y(x_0 + h) = y(x_0) + \int_{x_0}^{x_0+h} f(x, y(x)) dx$$

- オイラー法
$$\int_{x_0}^{x_0+h} f(x, y(x)) dx \rightarrow h f(x_0, y(x_0))$$

- 台形で近似できればもっと良い

$$\int_{x_0}^{x_0+h} f(x, y(x)) dx$$
$$\rightarrow \frac{h}{2} (f(x_0, y(x_0)) + f(x_0 + h, y(x_0 + h)))$$



(オイラー法) 精度の向上

予想値をオイラー法で求め、台形の面積を計算する

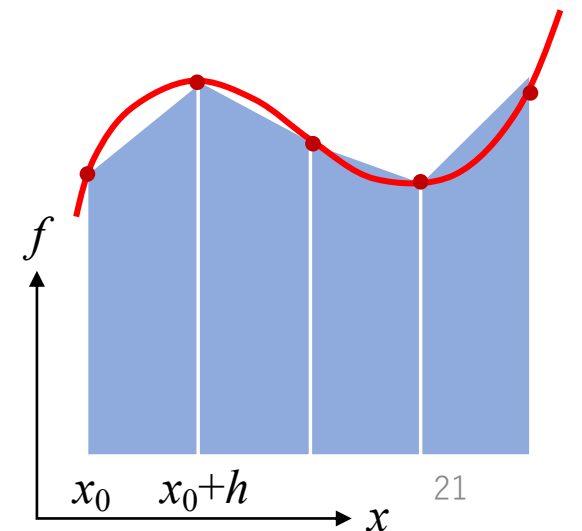
$$\overline{y(x_0 + h)} = y(x_0) + hf(x_0, y(x_0)) \quad \leftarrow y(x_0 + h) \text{ の予想値 (オイラー法)}$$

$$y(x_0 + h) = y(x_0) + \frac{h}{2} \left(f(x_0, y(x_0)) + f(x_0 + h, \overline{y(x_0 + h)}) \right)$$

テイラー展開と比較すると (各自やってみましょう)

- 1ステップ(h)あたりの誤差 $\rightarrow O(h^3)$
- 全ステップで発生する誤差 $\rightarrow O(h^2)$

「修正オイラー法」、精度は二次 $O(h^2)$



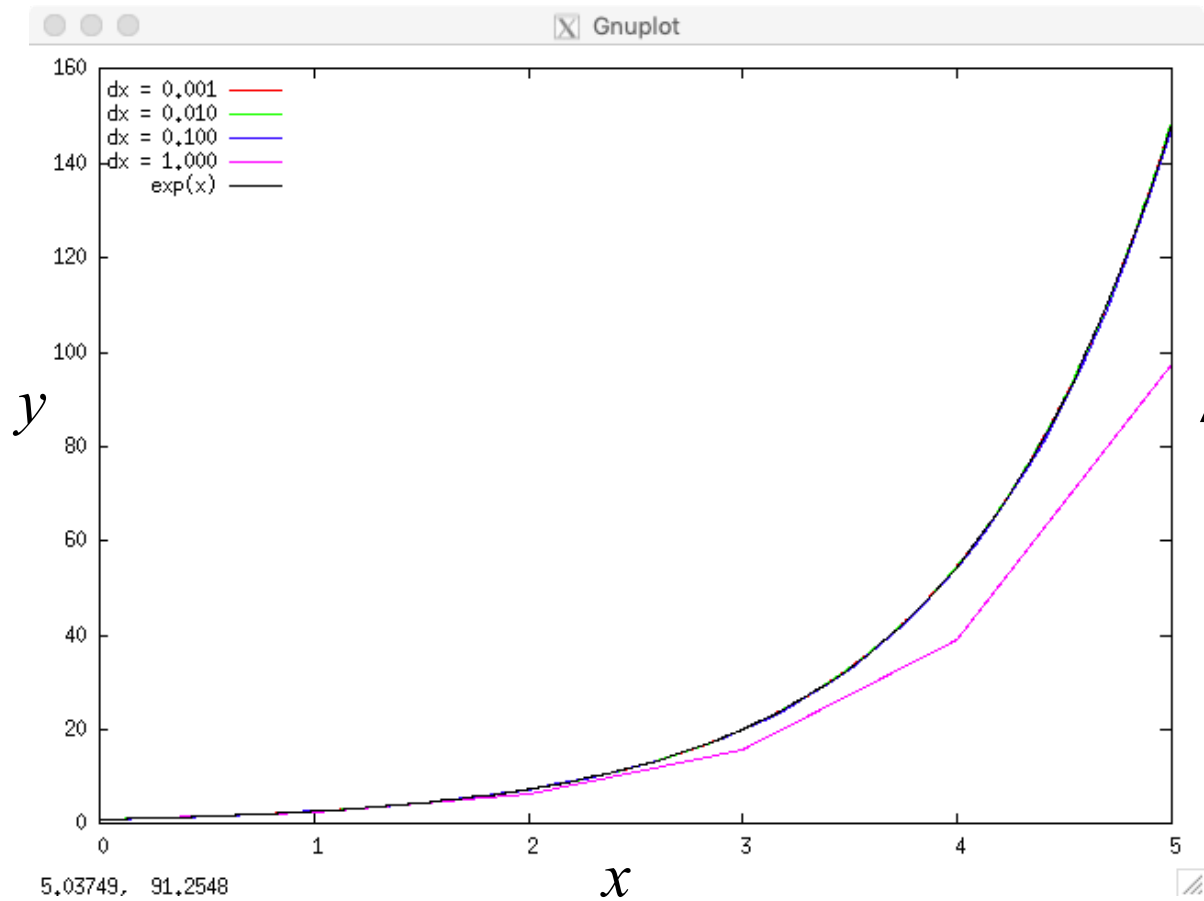
(オイラー法) 精度の向上

```
BEGIN{
  #dx   = 0.10  # from the command line
  x0    = 0.0   # initial value
  y0    = 1.0   # initial value
  xmax  = 10.0
  printf("%e %e %e\n", x0, y0, 0)
}

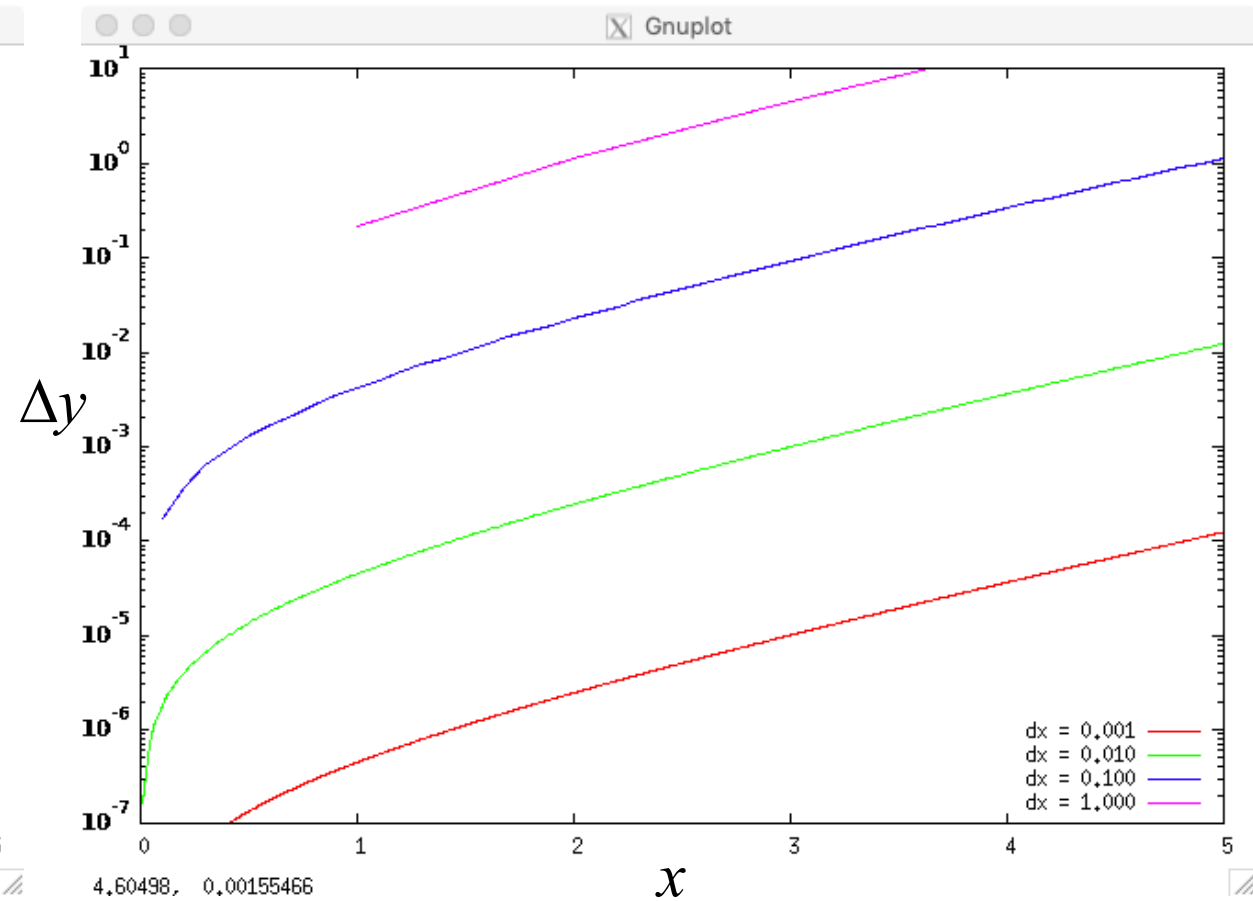
{
  x = x0
  y = y0
  while (x < xmax){
    yeu = y + fxy(x,y)*dx           # Euler
    y  += dx * (fxy(x,y) + fxy(x+dx,yeu))/2 # Trapezoid
    x  += dx
    printf("%e %e %e\n", x, y, fabs(y-yas(x)))
  }
}
```

(オイラー法) 精度の向上

$y = e^x$ との比較



$y = e^x$ との差分



(オイラー法) 精度の向上

- 公式は他にもある
- 一般論もある

<http://maya.phys.kyushu-u.ac.jp/~knomura/education/numerical-physics/text5/node2.html> から引用

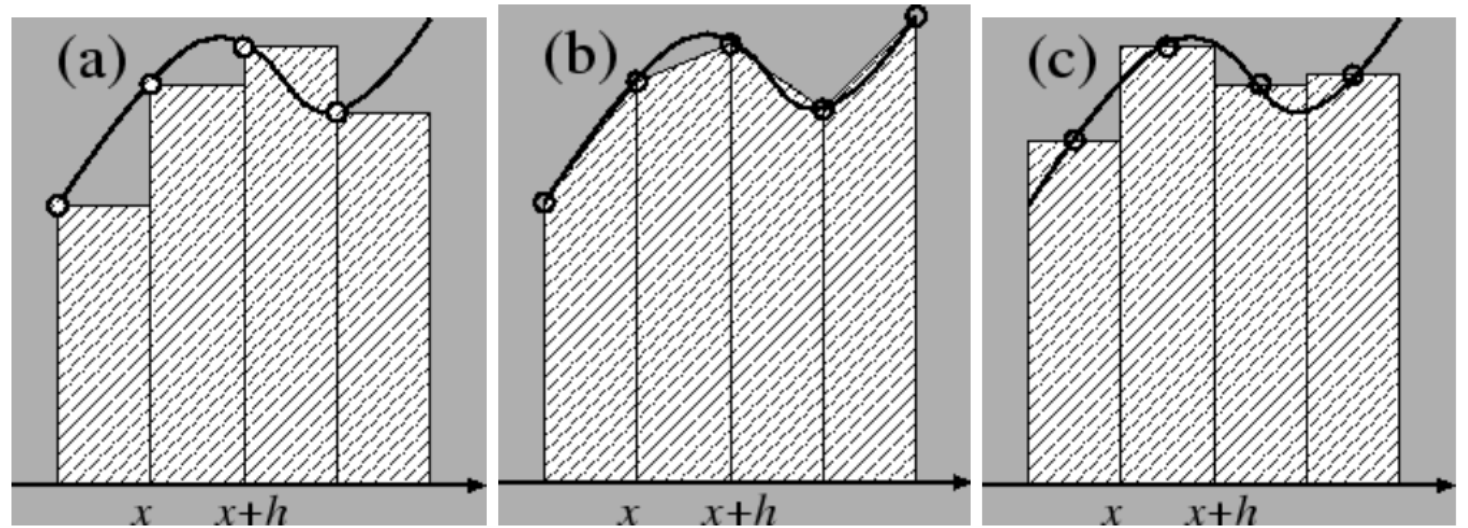


図: (a) オイラー法 (b) 修正オイラー法 (c) 2次のルンゲクッタ法に相当。

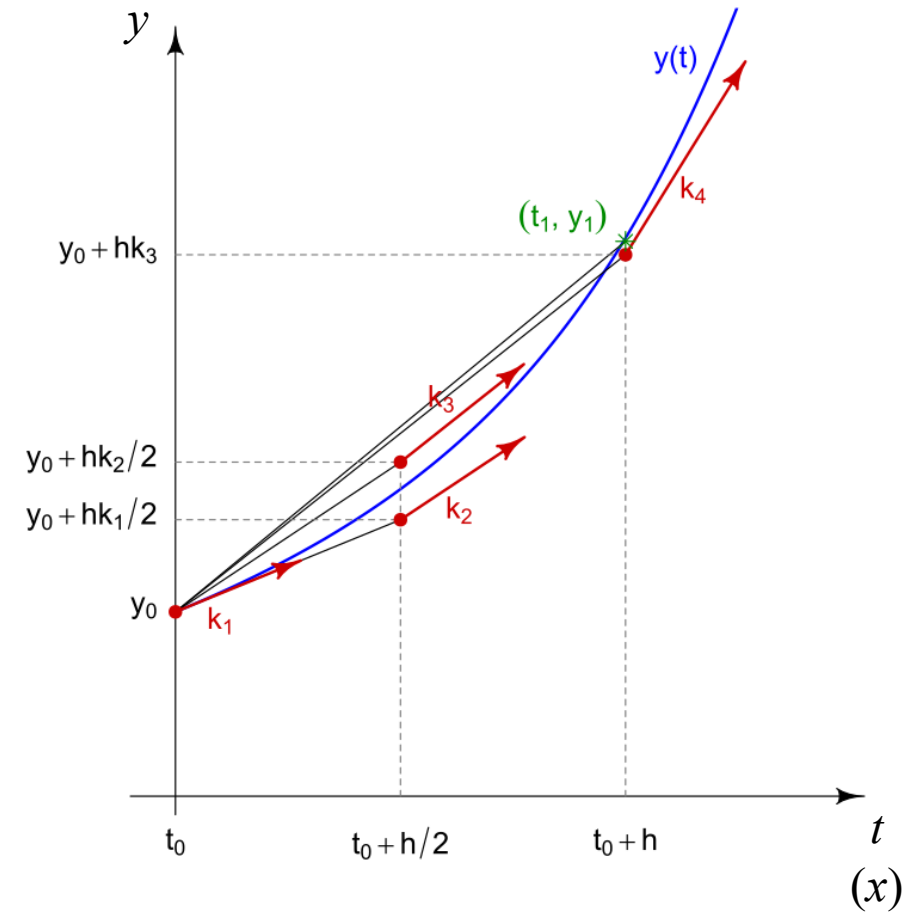
- 誤差を更に小さくしたい → 高次化

ルンゲ・クッタ(4次)

• 4段

$$\left\{ \begin{aligned} k_1 &= f(x_0, y(x_0)), \\ k_2 &= f\left(x_0 + \frac{h}{2}, y(x_0) + h\frac{k_1}{2}\right), \\ k_3 &= f\left(x_0 + \frac{h}{2}, y(x_0) + h\frac{k_2}{2}\right), \\ k_4 &= f(x_0 + h, y(x_0) + hk_3), \end{aligned} \right.$$

$$y(x_0 + h) = y(x_0) + \frac{h}{6} (k_1 + 2k_2 + 2k_3 + k_4)$$



- 4次の精度 $O(h^4)$
- 段数 = step

一般化の例 (4段)

$$d_0 = hf(x_0, y(x_0)),$$

$$d_1 = hf(x_0 + \alpha_1 h, y(x_0) + \beta_{10} d_0),$$

$$d_2 = hf(x_0 + \alpha_2 h, y(x_0) + \beta_{20} d_0 + \beta_{21} d_1),$$

$$d_3 = hf(x_0 + \alpha_3 h, y(x_0) + \beta_{30} d_0 + \beta_{31} d_1 + \beta_{32} d_2),$$

$$y(x_0 + h) = y(x_0) + \gamma_0 d_0 + \gamma_1 d_1 + \gamma_2 d_2 + \gamma_3 d_3$$

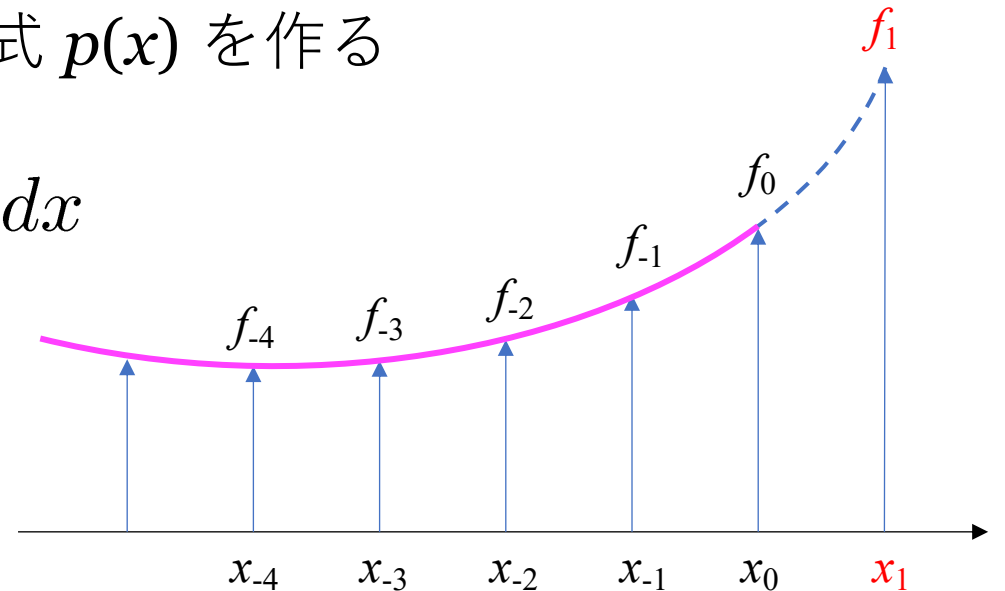
- これを $y(x_0+h)$ のテイラー展開と比較する
- h^4 項まで一致するように $\alpha_1, \alpha_2, \alpha_3, \beta_{10}, \dots, \gamma_0, \dots, \gamma_3$ を定める
 - 変数の数 > 方程式の数
 - 公式は唯一には決まらない

多段法（考え方のみ）

- n段(n回の関数 f 計算)が必要な理由
 - n次の多項式を作ってテイラー展開を近似する
- 過去に計算した結果を使えば良いのでは？
 - 例. $f_0, f_{-1}, f_{-2}, \dots$ を使いn次の近似多項式 $p(x)$ を作る

$$y(x_0 + h) = y(x_0) + \int_x^{x_0+h} p(x) dx$$

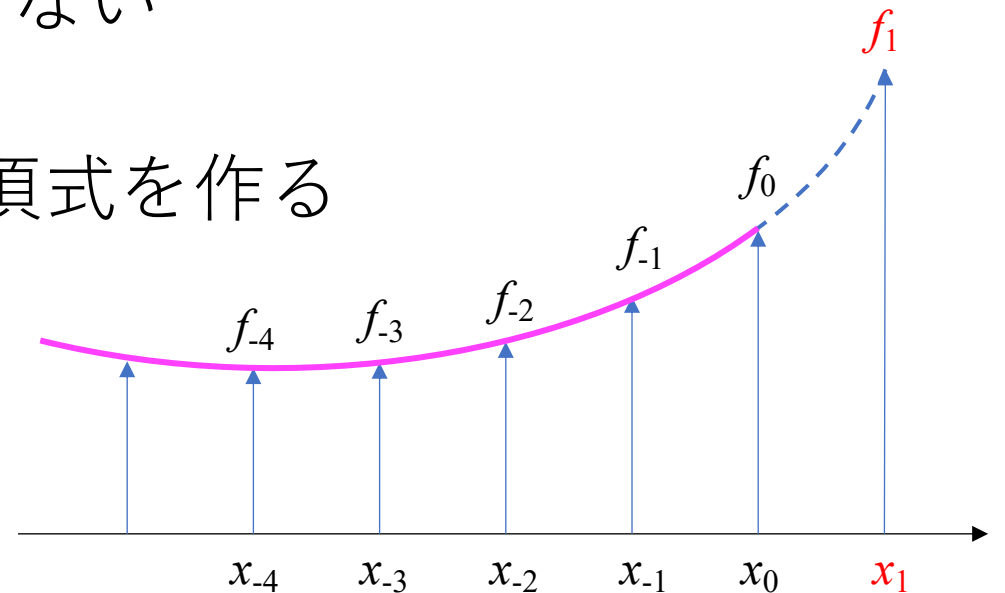
- 段数 = step



多段法（考え方のみ）

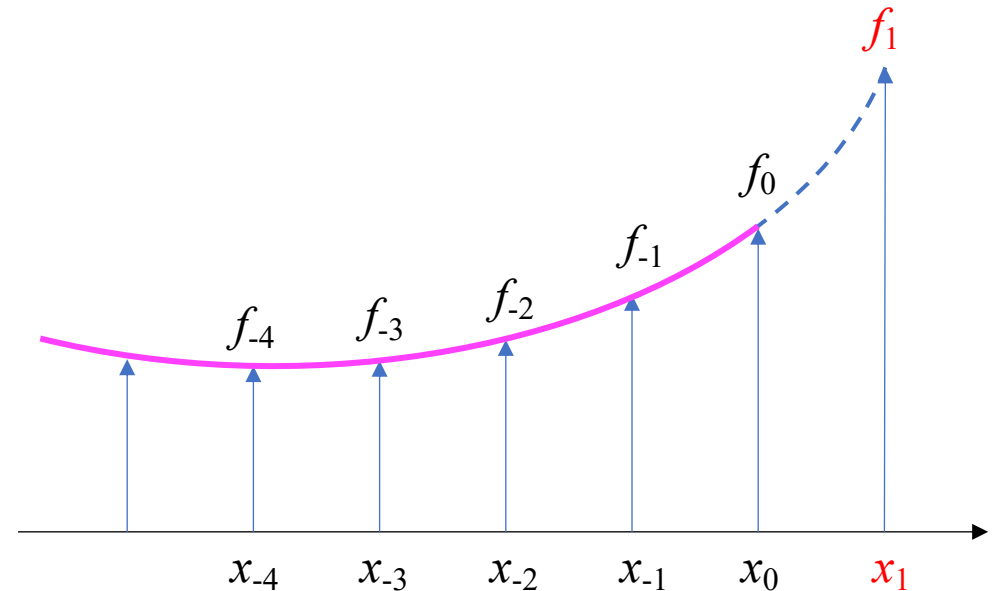
- 滑らかな関数であれば高い精度を得られる
- だが、出発値を揃えるのが面倒
- 単なる外挿では安定した積分ができない
 - 例. Adams-Bashforth公式
- 対策: $f_1, f_0, f_{-1}, f_{-2}, \dots$ を使い近似多項式を作る
 - 何らかの予測 → 補間 → 修正
 - 例. Adams-Moulton公式

「予測子・修正子法」



多段法（考え方のみ）

- が、出発値の準備の面倒さは避け得ない
- 過去の値ではなく、高階の導関数を使って多項式近似しては？
 - 例. エルミート法 [Makino \(1991\)](#) など
 - 詳しい解説 → 小久保先生の講義



やや唐突だが、二階の常微分方程式

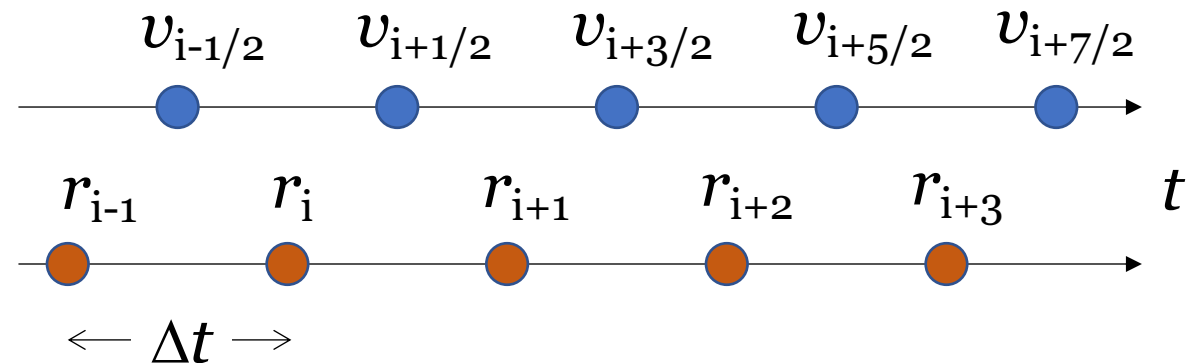
$$\frac{d^2 \mathbf{r}}{dt^2} = F(\mathbf{r}), \text{つまり } \frac{d\mathbf{v}}{dt} = F(\mathbf{r}), \frac{d\mathbf{r}}{dt} = \mathbf{v}$$

に関し、以下の方法は良い。

$$\begin{cases} \mathbf{v}_{i+\frac{1}{2}} = \mathbf{v}_{i-\frac{1}{2}} + F(\mathbf{r}_i)\Delta t, \\ \mathbf{r}_{i+1} = \mathbf{r}_i + \mathbf{v}_{i+\frac{1}{2}}\Delta t \end{cases}$$

出発時と終了時は以下

$$\begin{cases} \mathbf{v}_{\frac{1}{2}} = \mathbf{v}_0 + F(\mathbf{r}_0)\frac{\Delta t}{2}, \\ \mathbf{v}_n = \mathbf{v}_{n-\frac{1}{2}} + F(\mathbf{r}_n)\frac{\Delta t}{2} \end{cases}$$



やや唐突だが、二階の常微分方程式

- 様々な記法がある
- 名称も様々なものがある
 - [Leapfrog \(LF\)](#)
 - Velocity Verlet法
 - ...
- この方法の正体
 - 話の後半で述べる
- 実装と利用
 - 実習 + 小久保先生の講義

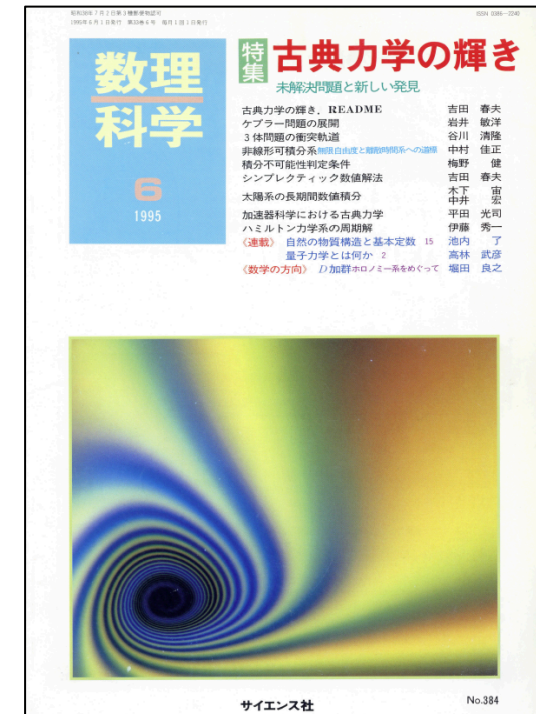
$$\left\{ \begin{array}{l} \mathbf{v}_{i+\frac{1}{2}} = \mathbf{v}_i + F(\mathbf{r}_i) \frac{\Delta t}{2}, \\ \mathbf{r}_{i+1} = \mathbf{r}_i + \mathbf{v}_{i+\frac{1}{2}} \Delta t, \\ \mathbf{v}_{i+1} = \mathbf{v}_{i+\frac{1}{2}} + F(\mathbf{r}_{i+1}) \frac{\Delta t}{2} \end{array} \right.$$

正準形式の解法

正準形式の解法

- 1990年代から急速に研究が盛んになった
- 解析力学の理論体系に裏付けされており天体力学と親和性が高い

- この講義(後半部)が沿う解説記事
 - 膨大な数の文献がある。日本語でも



復習

- (重力)N体問題 → 多くは正準形式で記述できる
- 一般化座標 q , 一般化運動量 p
- 正準形の運動方程式

$$\frac{dq_j}{dt} = \frac{\partial H}{\partial p_j}, \quad \frac{dp_j}{dt} = -\frac{\partial H}{\partial q_j} \quad (j = 1 \dots n)$$

- ハミルトニアン

$$H = \text{運動エネルギー}(p) + \text{ポテンシャルエネルギー}(q)$$

復習

- 正準変換：変数変換 $(q, p) \rightarrow (Q, P)$ 後も正準形式に書けること

$$\frac{dQ_j}{dt} = \frac{\partial \tilde{H}}{\partial P_j}, \quad \frac{dP_j}{dt} = -\frac{\partial \tilde{H}}{\partial Q_j} \quad (j = 1 \dots n)$$

- ポアソン括弧 $\{, \}$

$$\{A, B\}_{(q,p)} = \sum_{i=1}^n \left(\frac{\partial A}{\partial q_i} \frac{\partial B}{\partial p_i} - \frac{\partial B}{\partial q_i} \frac{\partial A}{\partial p_i} \right)$$

復習

- ハミルトン系の時間発展
$$\begin{pmatrix} \mathbf{q}(0) \\ \mathbf{p}(0) \end{pmatrix} \rightarrow \begin{pmatrix} \mathbf{q}(\tau) \\ \mathbf{p}(\tau) \end{pmatrix}$$
- これを座標変換と見做す \rightarrow 正準変換になっている
 - 外積による表現 ($\Sigma dq \wedge dp$)
 - ポアソン括弧による表現
 - 行列による表現
 - \vdots
- この性質の保持を主目的とした数値解法がある

形式的な解

- ポアソン括弧演算子 $\{, G\}$

$$\{, G\} F \equiv \{F, G\}$$

形式的な解

- $z = q$ または p , ポアソン括弧 $\{, \}$ を使うと正準方程式は

$$\frac{dz}{dt} = \{z, H(z)\}$$

- これを以下と見做す

$$\frac{dz}{dt} = \{, H\} z$$

- 形式的な解

$$z(\tau) = e^{\tau\{, H\}} z(0)$$

形式的な解

- ハミルトニアン分割 (運動エネルギー T , ポテンシャル V)

$$H = T(p) + V(q)$$

- 形式的な解
$$\begin{aligned} z(\tau) &= e^{\tau\{, H\}} z(0) \\ &= e^{\tau\{, T+V\}} z(0) \\ &= e^{(\tau\{, T\} + \tau\{, V\})} z(0) \\ &= e^{\tau(A+B)} z(0) \end{aligned}$$

$$A \equiv \{, T\}, \quad B \equiv \{, V\}$$

形式的な解

- 以下の操作(演算)では厳密な解が求まる

$$z(\tau) = e^{\tau A} z(0), \quad z(\tau) = e^{\tau B} z(0)$$

- 例. $e^{\tau A}$ の式を通常 of 形に書き下す

$$\frac{dq}{dt} = \frac{\partial T(p)}{\partial p}, \quad \frac{dp}{dt} = -\frac{\partial T(p)}{\partial q}$$

- $p = \text{定数}, q \propto t \rightarrow q$ 方向の等速直線運動
- $e^{\tau B}$ の式も同様
 - $q = \text{定数}, p \propto t \rightarrow p$ 方向の等速直線運動
- これらの操作(演算)をうまく組み合わせたい

形式的な解

- 真の時間発展 $e^{\tau(A+B)}$ を形式的に展開

$$e^{\tau(A+B)} = 1 + \tau(A+B) + \frac{\tau^2}{2} (A+B)^2 + O(\tau^3)$$

$$= 1 + \tau(A+B) + \frac{\tau^2}{2} (A^2 + AB + BA + B^2) + O(\tau^3)$$

- (実際に計算できる) $e^{\tau A} e^{\tau B}$ は

$$e^{\tau A} e^{\tau B} = \left(1 + \tau A + \frac{\tau^2}{2} A^2 + O(\tau^3) \right) \left(1 + \tau B + \frac{\tau^2}{2} B^2 + O(\tau^3) \right)$$

$$= 1 + \tau(A+B) + \frac{\tau^2}{2} (A^2 + 2AB + B^2) + O(\tau^3)$$

- $O(\tau)$ まで合致

一次のスキーム

- $e^{\tau A} e^{\tau B}$ を通常形式で書き下す

$$q(\tau) = q(0) + \tau \left. \frac{\partial T}{\partial p} \right|_{p(0)}$$

$$p(\tau) = p(0) - \tau \left. \frac{\partial V}{\partial q} \right|_{q(\tau)}$$

シンプレクティック数値解法
(一次)

一次のスキーム

- オイラー法ももちろん適用できる

$$q(\tau) = q(0) + \tau \left. \frac{\partial T}{\partial p} \right|_{p(0)}$$

$$p(\tau) = p(0) - \tau \left. \frac{\partial V}{\partial q} \right|_{q(0)}$$

調和振動子（オイラー法）

```
BEGIN{
  #dt =          0.10 # from the command line
  q  = q_pre = 1.0  # initial value
  p  = p_pre = 0.0  # initial value
  tmax = 100.0
}
{
  t = 0
  while (t < tmax){
    t += dt
    q += dt*p_pre
    p -= dt*q_pre
    printf("%e %e %e\n", t, q, p)
    q_pre = q
    p_pre = p
  }
}
```

ハミルトニアン

$$H = \frac{1}{2} (q^2 + p^2)$$

運動方程式

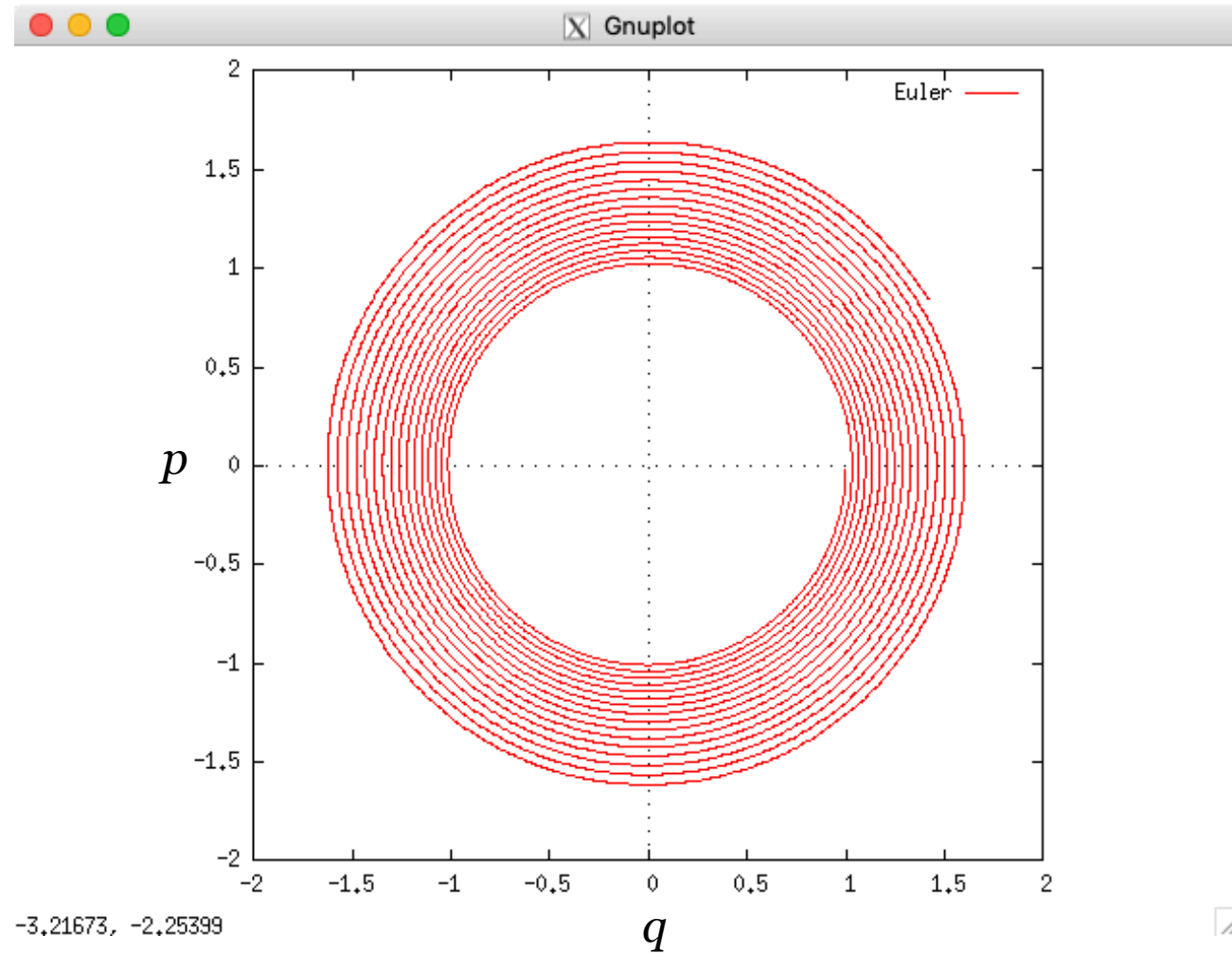
$$\frac{dq}{dt} = \frac{\partial H}{\partial p} = p, \quad \frac{dp}{dt} = -\frac{\partial H}{\partial q} = -q$$

厳密解（回転）

$$\begin{pmatrix} q(\tau) \\ p(\tau) \end{pmatrix} = \begin{pmatrix} \cos \tau & \sin \tau \\ -\sin \tau & \cos \tau \end{pmatrix} \begin{pmatrix} q(0) \\ p(0) \end{pmatrix}$$

調和振動子（オイラー法）

- 発散する



調和振動子（シンプレクティック数値解法）

```
BEGIN{
  #dt =          0.10 # from the command line
  q  =          1.0 # initial value
  p  =          0.0 # initial value
  tmax      = 100.0
}
{
  t = 0
  while (t < tmax){
    t += dt
    q += dt*p
    p -= dt*q
    printf("%e %e %e\n", t, q, p)
  }
}
```

ハミルトニアン

$$H = \frac{1}{2} (q^2 + p^2)$$

運動方程式

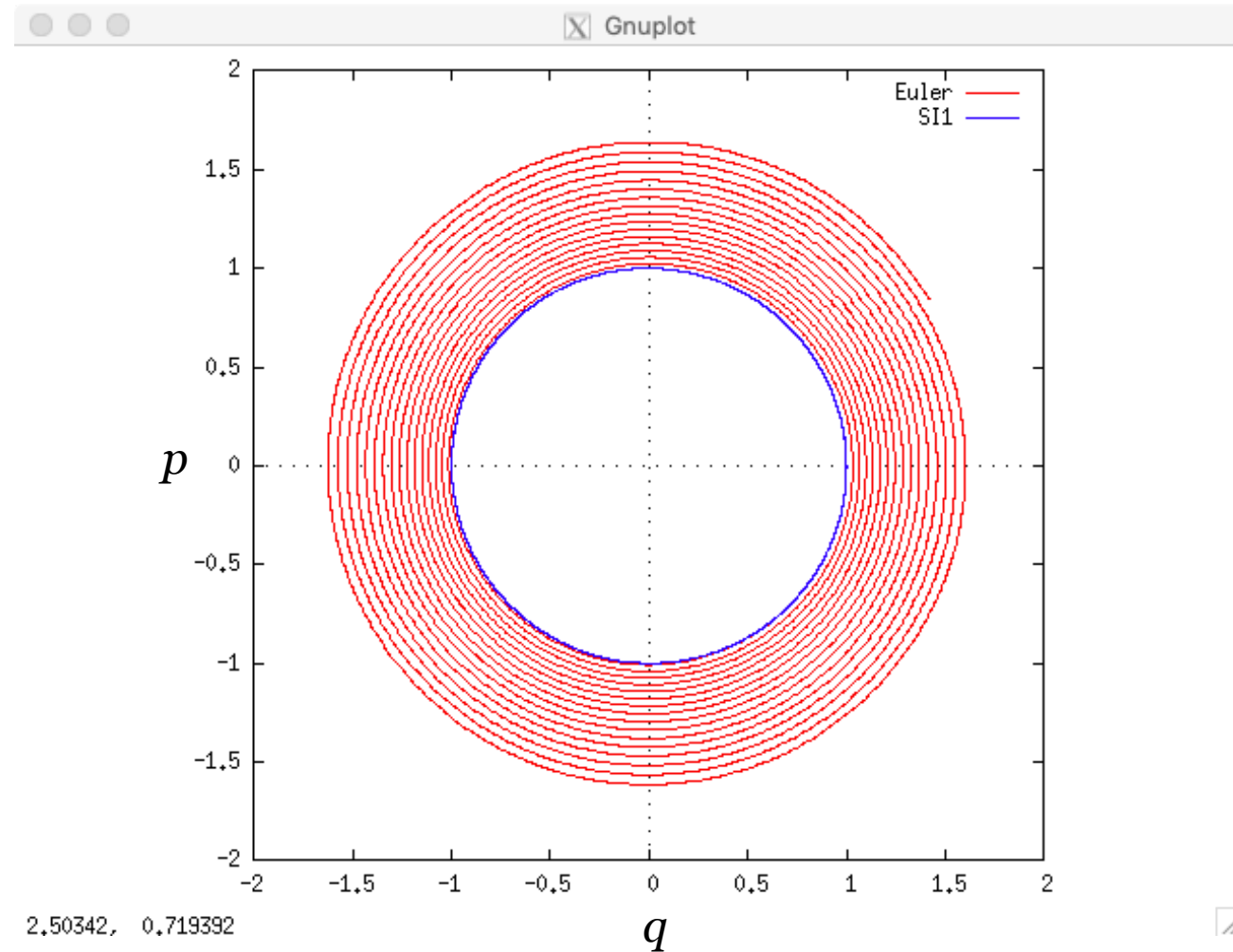
$$\frac{dq}{dt} = \frac{\partial H}{\partial p} = p, \quad \frac{dp}{dt} = -\frac{\partial H}{\partial q} = -q$$

厳密解（回転）

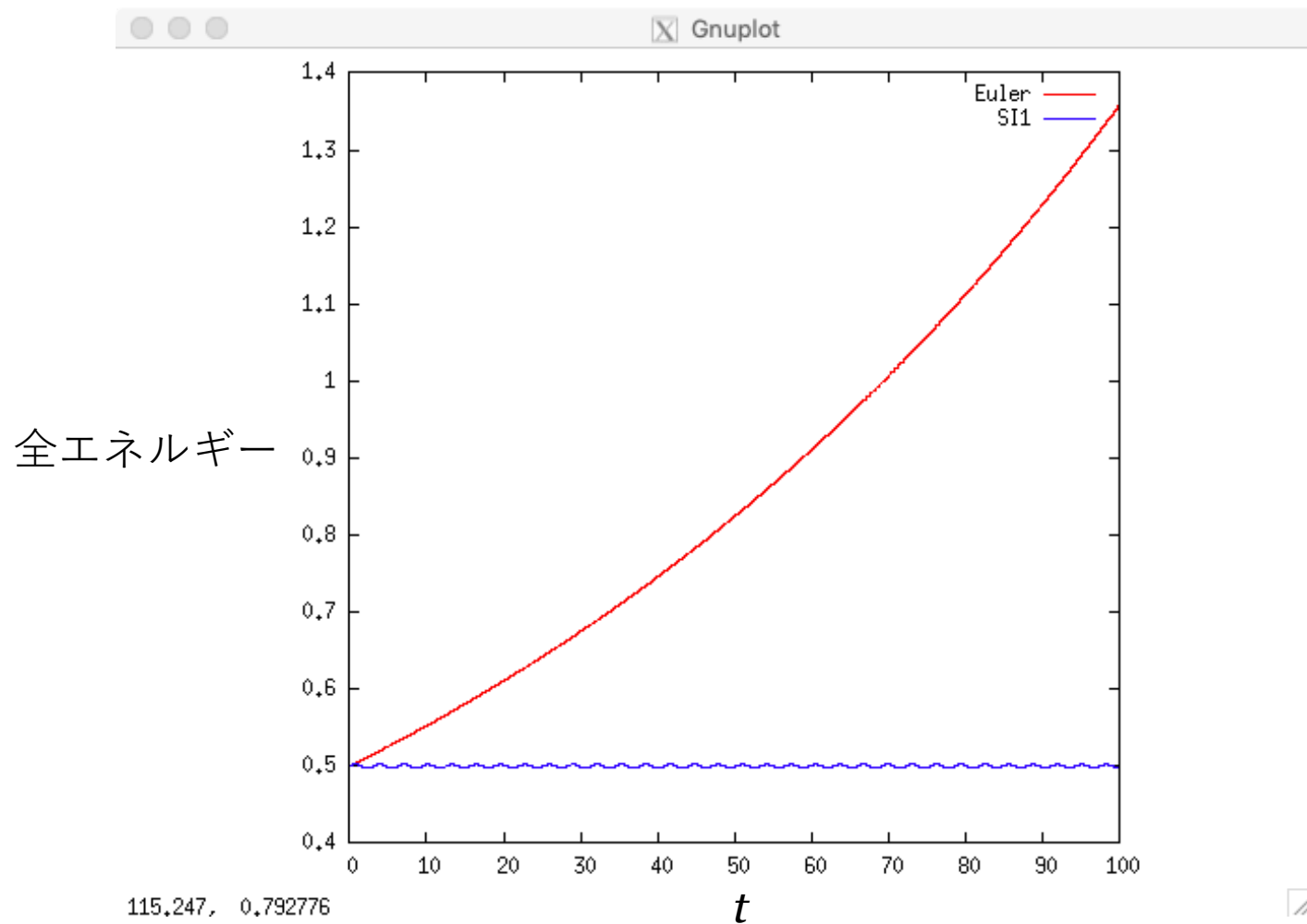
$$\begin{pmatrix} q(\tau) \\ p(\tau) \end{pmatrix} = \begin{pmatrix} \cos \tau & \sin \tau \\ -\sin \tau & \cos \tau \end{pmatrix} \begin{pmatrix} q(0) \\ p(0) \end{pmatrix}$$

調和振動子（シンプレクティック数値解法）

- 発散しない



全エネルギーの保存状態



違いは少しだけ

オイラー法のコードで以下の部分

```
while (t < tmax){  
    q += dt*p_pre  
    p -= dt*q_pre  
}
```

を書き誤り、以下

```
while (t < tmax){  
    q += dt*p  
    p -= dt*q  
}
```

のようになるとシンプレクティック数値解法になる

理論的には

- 計算スキーム(左辺)、それが支配する系(右辺)

$$e^{\tau\{, T\}} e^{\tau\{, V\}} = e^{\tau\{, \tilde{H}\}}$$

- ハミルトニアン \tilde{H} とは？

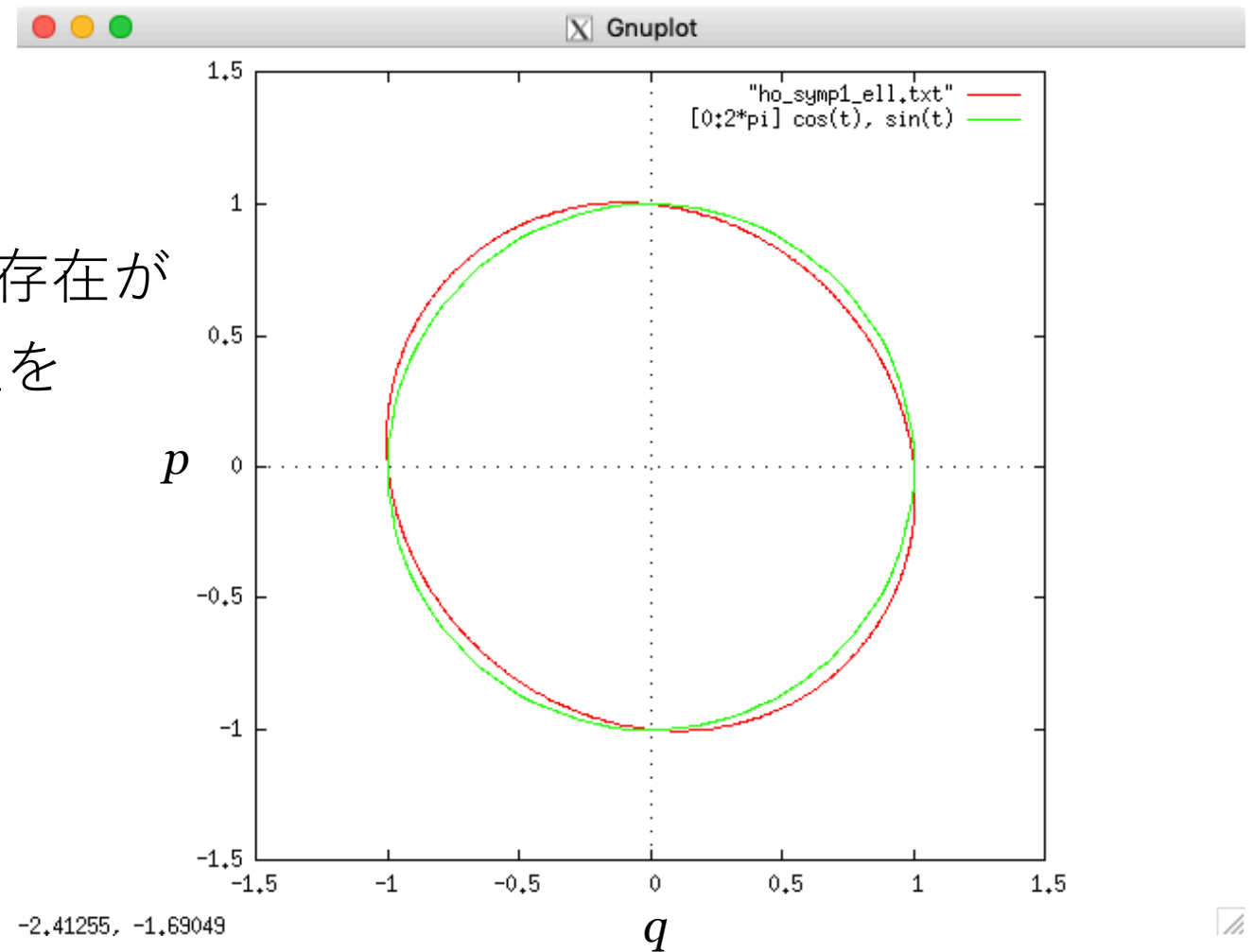
$$\begin{aligned} \tilde{H} = & T + V + \frac{\tau}{2}\{T, V\} \\ & + \frac{\tau^2}{12} (\{\{T, V\}, V\} + \{\{V, T\}, T\}) + O(\tau^3) \end{aligned}$$

- 元々のハミルトニアンとの差異は $O(\tau)$ に留まる

理論的には

【注】

保存量(代理ハミルトニアン)の存在が
この方法のシンプレクティック性を
示すわけではない



精度の向上

- 高次スキームの一般形

$$e^{\tau(A+B)} = e^{c_1\tau A} e^{d_1\tau B} \times \dots \times e^{c_k\tau A} e^{d_k\tau B} + O(\tau^{n+1})$$

- 形式的な展開を代入

$$e^{\tau X} = 1 + \frac{X}{1!} + \frac{X^2}{2!} + \frac{X^3}{3!} + \dots$$

- $A, B, A^2, AB, BA, B^2, \dots$ の係数を比較し、未知数 $c_1, c_2, \dots, d_1, d_2, \dots$ (と k) を定める

係数の計算例 (一次)

```
# [A,B] != [B,A]
with(Physics):
Setup(noncommutativeprefix = {A, B}):

# exp(x) up to x^1
expx := x -> 1 + x;

# the authentic operator (1st)
expx := x -> 1 + x;

# the authentic operator (1st)
opeauth := expand(expx(t*(A+B)));

# si operator (1st)
opesim1 := convert(expand( expx(c1*t*(A))*expx(d1*t*(B)) ),polynom);

# sort and truncate up to t^2
sort(mtaylor(combine(opeauth), [t], 2), t, ascending);
sort(mtaylor(combine(opesim1), [t], 2), t, ascending);
```

係数の計算例 (一次)

```
|¥^/|      Maple 2018 (X86 64 LINUX)
. _|¥|    |/_|. Copyright (c) Maplesoft, a division of Waterloo Maple Inc. 2018
¥  MAPLE / All rights reserved. Maple is a trademark of
< _____ > Waterloo Maple Inc.
      |
      Type ? for help.
```

```
> expx := x -> 1 + x;
```

```
      expx := x -> x + 1
```

```
> opeauth := expand(expx(t*(A+B)));
```

```
      opeauth := A t + B t + 1
```

```
> opesim1 := convert(expand(expx(c1*t*(A))*expx(d1*t*(B)) ),polynom);
```

```
      2
```

```
      opesim1 := c1 t  d1 (A B) + c1 t A + d1 t B + 1
```

```
> sort(mtaylor(combine(opeauth), [t], 2), t, ascending);
```

```
      1 + (A + B) t
```

```
> sort(mtaylor(combine(opesim1), [t], 2), t, ascending);
```

```
      1 + (A c1 + B d1) t
```

係数の計算例（一次）

$$c_1 = 1, d_1 = 1 (k = 1)$$

係数の計算例 (二次)

```
# [A,B] != [B,A]
with(Physics):
Setup(noncommutativeprefix = {A, B}):

# exp(x) up to x^2
expx := x -> 1 + x + (x^2)/2!;

# the authentic operator (2nd)
opeauth := expand(expx(t*(A+B)));

# si operator (2nd)
opesim2 := convert(expand( expx(c1*t*(A))*expx(d1*t*(B))
                    *expx(c2*t*(A))*expx(d2*t*(B)) ),polynom);

# sort and truncate up to t^2
sort(mtaylor(combine(opeauth), [t], 3), t, ascending);
sort(mtaylor(combine(opesim2), [t], 3), t, ascending);
```


係数の計算例 (二次)

```
> expx := x -> 1 + x + (x^2)/2!;
```

```
expx := x -> 1 + x + Physics:-`*`(Physics:-`^^`(x, 2), Physics:-`^^`(2!, -1))
```

```
> opeauth := expand(expx(t*(A+B)));
```

```
opeauth := 1 + t A + t B + 1/2 t 2^(A, 2) + 1/2 t (A B) + 1/2 t (B A) + 1/2 t 2^(B, 2)
```

(中略)

```
> sort(mtaylor(combine(opeauth), [t], 3), t, ascending);
```

```
1 + (A + B) t + 1/2 (2^(A, 2) + A B + B A + 2^(B, 2)) t
```

```
> sort(mtaylor(combine(opesim2), [t], 3), t, ascending);
```

```
1 + (A c1 + A c2 + B d1 + B d2) t + 1/2 (c1 2^(A, 2) + 2 c2 c1 2^(A, 2)
```

```
+ c2 2^(A, 2) + 2^(B, 2) d1 + 2 2^(B, 2) d2 d1 + 2^(B, 2) d2
```

```
+ 2 d1 c2 (B A) + 2 c1 d1 (A B) + 2 c1 d2 (A B) + 2 c2 d2 (A B)) t
```

係数の計算例 (二次)

$$c_1 + c_2 = 1$$

$$d_1 + d_2 = 1$$

$$2(c_1d_1 + c_1d_2 + c_2d_2) = 1$$

$$2c_2d_1 = 1$$

$$c_1^2 + 2c_1c_2 + c_2^2 = 1$$

$$d_1^2 + 2d_1d_2 + d_2^2 = 1$$

$$\rightarrow c_1 = c_2 = 1/2, d_1 = 1, d_2 = 0 (k = 2)$$

二次の方法

形式的に書けば

$$e^{\tau(A+B)} = e^{\frac{\tau}{2}A} e^{\tau B} e^{\frac{\tau}{2}A} + O(\tau^3)$$

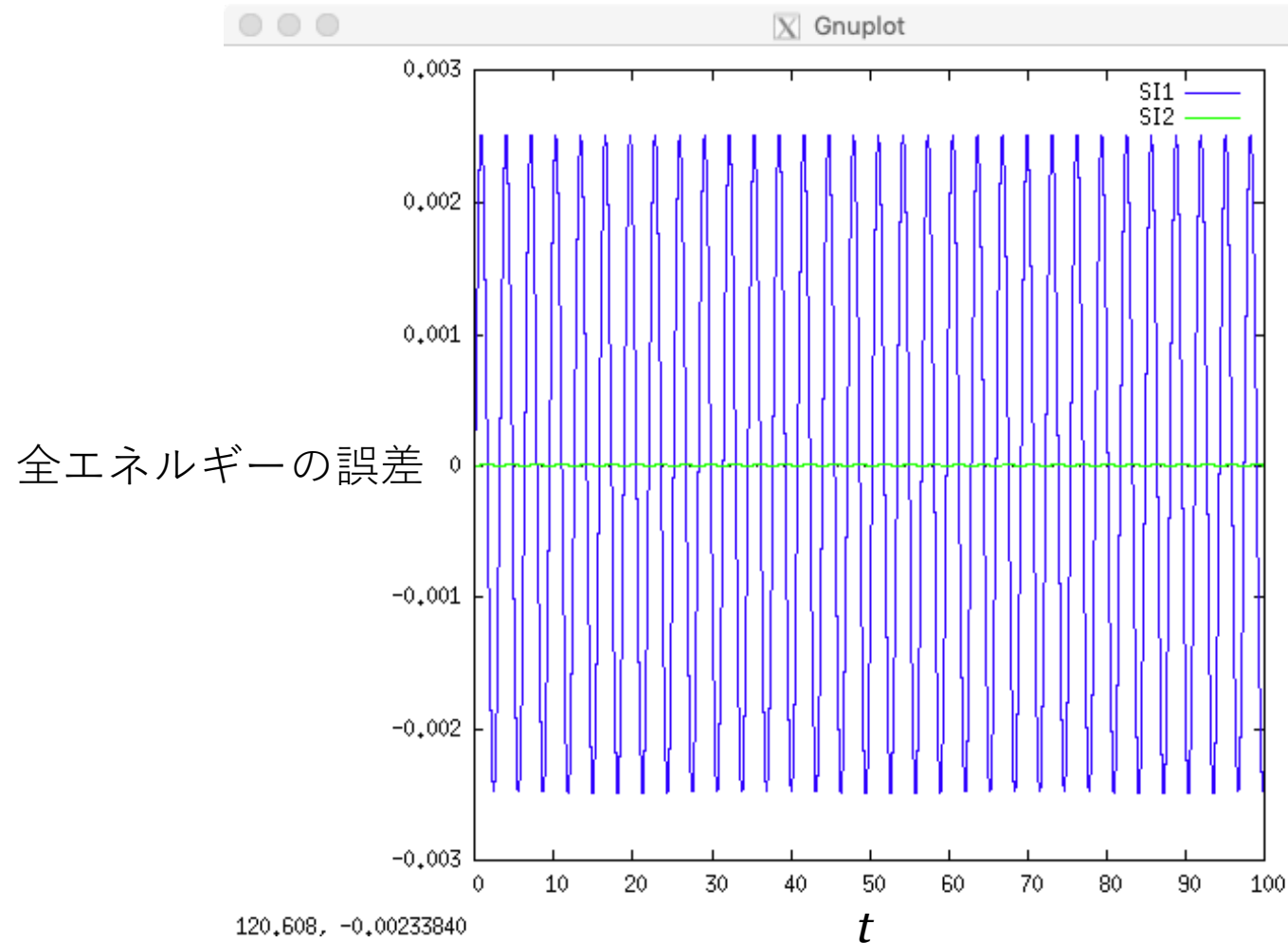
実際的な書き方

$$\left. \begin{aligned} q^* &= q(0) + \frac{\tau}{2} \frac{\partial T}{\partial p} \Big|_{p(0)} \\ p(\tau) &= p(0) - \tau \frac{\partial V}{\partial q} \Big|_{q^*} \\ q(\tau) &= q^* + \frac{\tau}{2} \frac{\partial V}{\partial q} \Big|_{p(\tau)} \end{aligned} \right\} \text{Leapfrogと等価}$$

二次の方法 (または Leapfrog)

```
BEGIN{
  #dt =          0.10 # from the command line
  q  =          1.0  # initial value
  p  =          0.0  # initial value
  tmax = 100.0
}
{
  t = 0
  while (t < tmax){
    t += dt
    qd = q  + dt*p/2
    p  = p  - dt*qd
    q  = qd + dt*p/2
    printf("%e %e %e\n", t, q, p)
  }
}
```

二次の方法（または Leapfrog）



一次
二次 (Leapfrog)

理論的には

- 計算スキーム(左辺)、それが支配する系(右辺)

$$e^{\frac{\tau}{2}\{, T\}} e^{\tau\{, V\}} e^{\frac{\tau}{2}\{, T\}} = e^{\tau\{, \tilde{H}\}}$$

- ハミルトニアン \tilde{H} とは？

$$\tilde{H} = T + V$$

$$+ \frac{\tau^2}{24} (2\{\{T, V\}, V\} - \{\{V, T\}, T\}) + O(\tau^4)$$

- 元々のハミルトニアンとの差異は $O(\tau^2)$ に留まる

高次の方法

- 四次の形式展開

$$e^{\tau A} e^{\tau B} = e^{c_1 \tau A} e^{d_1 \tau B} e^{c_2 \tau A} e^{d_2 \tau B} e^{c_3 \tau A} e^{d_3 \tau B} e^{c_4 \tau A} e^{d_4 \tau B} + O(\tau^5)$$

- 係数セットの例

$$c_1 = c_4 = \frac{1}{2 \left(2 - 2^{\frac{1}{3}} \right)}, \quad c_2 = c_3 = \frac{1 - 2^{\frac{1}{3}}}{2 \left(2 - 2^{\frac{1}{3}} \right)}$$

$$d_1 = d_3 = \frac{1}{2 - 2^{\frac{1}{3}}}, \quad d_2 = \frac{-2^{\frac{1}{3}}}{2 - 2^{\frac{1}{3}}}, \quad d_4 = 0$$

- 高次の解法 → 二次の方法を合成して構成可能 [Yoshida \(1990\)](#)

この方法の長所のまとめ

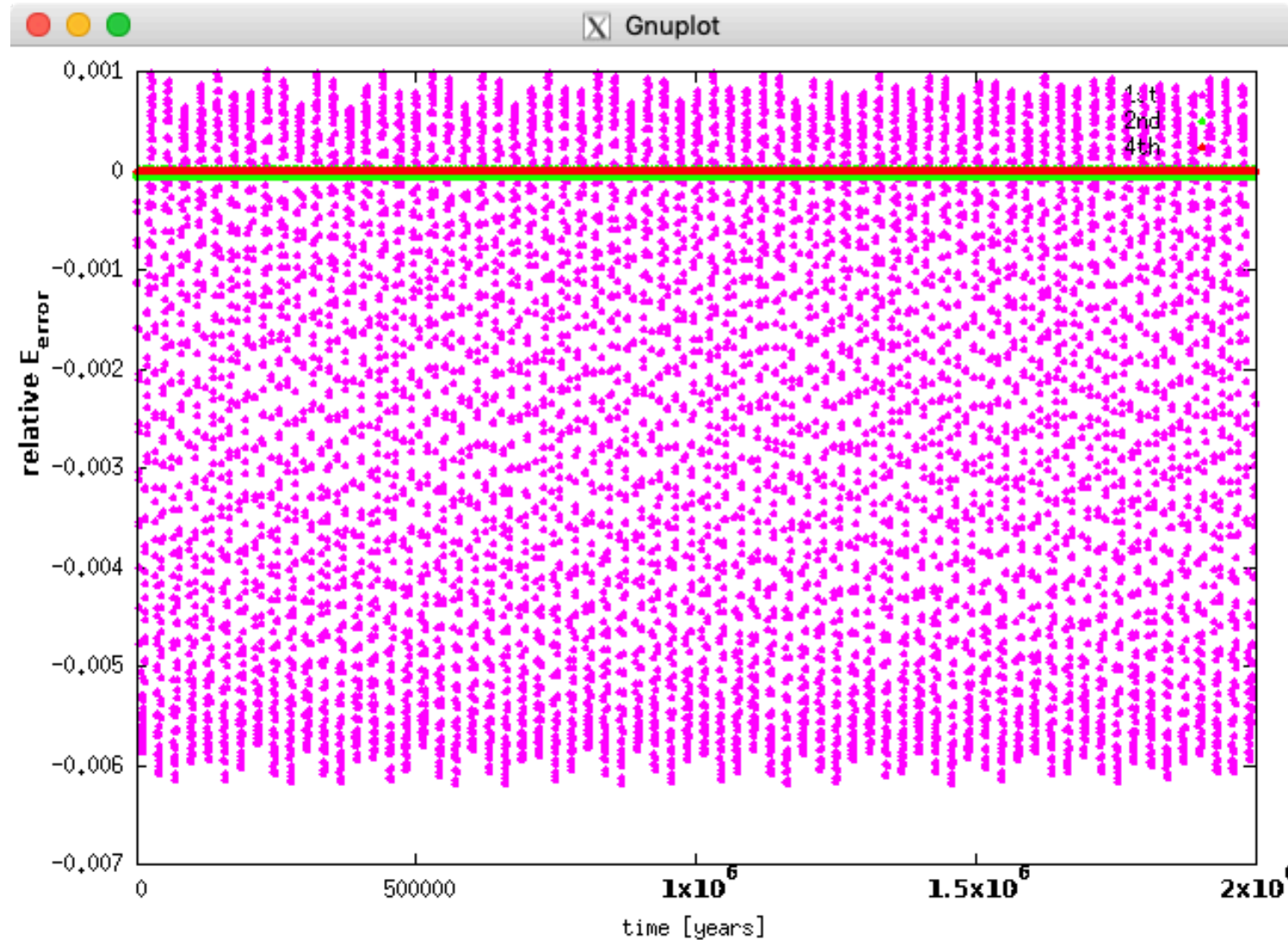
- 時間発展のシンプレクティック性が保たれる
- 全エネルギーの誤差が蓄積されない
- 偶数次の解法は時間反転に対して対称である
- $H = T(p) + V(q)$ 型ではN体系の全角運動量が厳密保存される
- 誤差評価や解法拡張に正準摂動論の知識体系を活用できる
 - [堀源一郎先生の方法](#) → 天体力学との親和性が高い

Yoshida (1990,
preprint)

【参考】元の微分方程式が持つ或る性質を厳密に保持する解法

- 「幾何学的積分法」 (geometric integration)

N体の計算例 (N=6, 太陽+木星土天海冥)

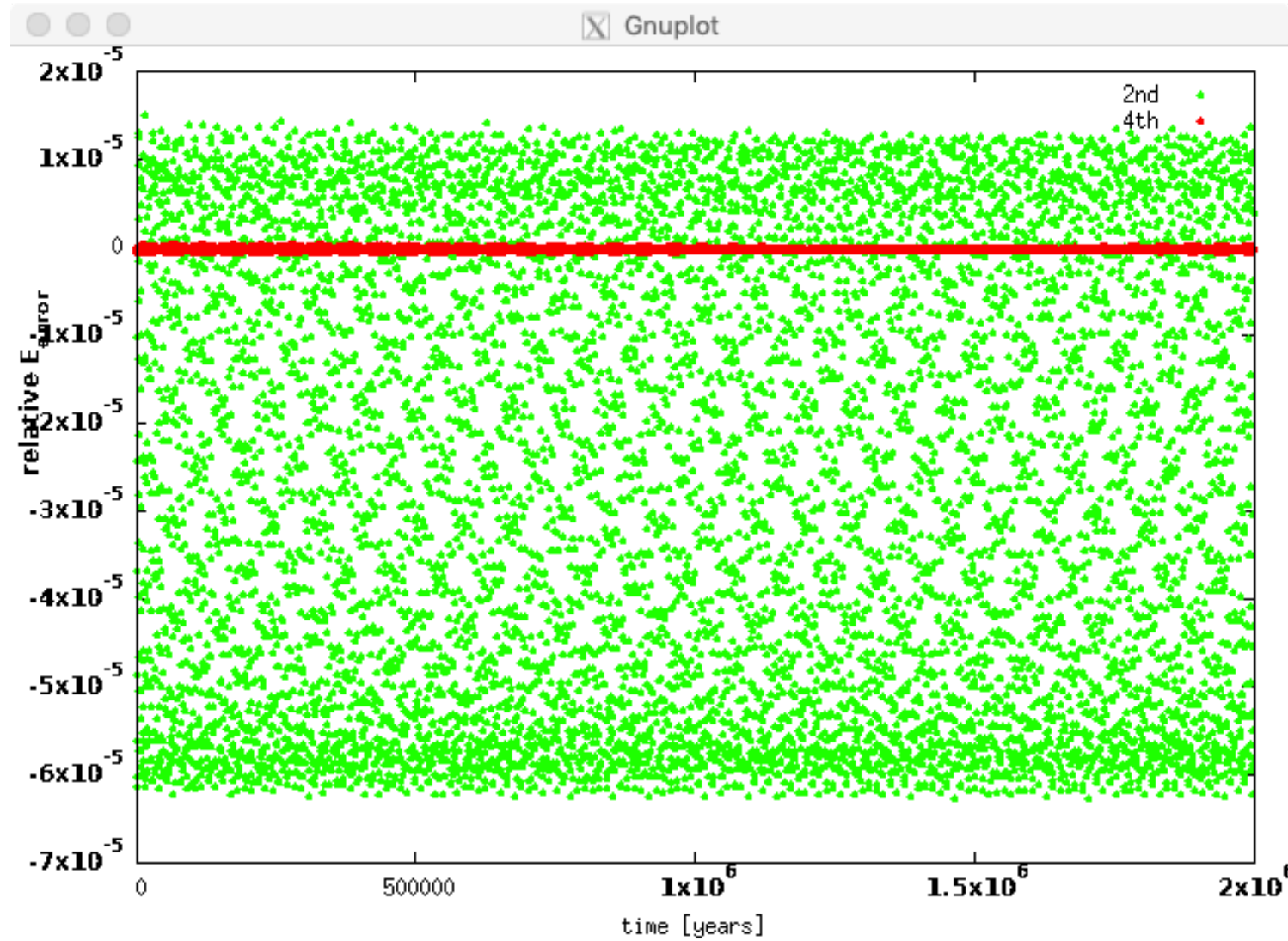


全エネルギーの
相対誤差

- 1次
- 2次 (LF)
- 4次

誤差の平均値の
0からのずれが
重要

N体の計算例 (N=6, 太陽+木星土天海冥)

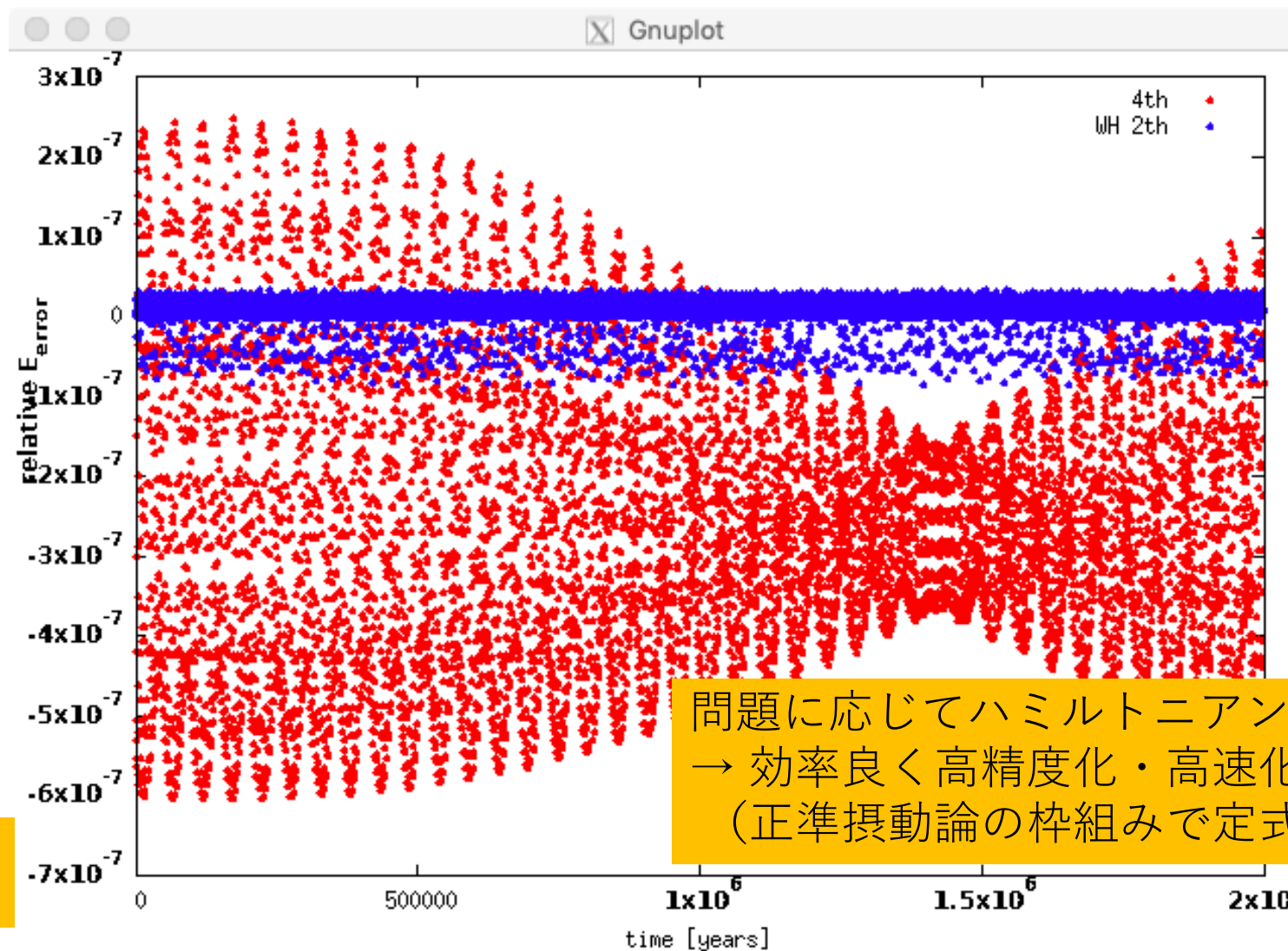


全エネルギーの
相対誤差

- 2次 (LF)
- 4次

時刻

N体の計算例 (N=6, 太陽+木星土天海冥)



全エネルギーの
相対誤差

● 4次
● 2次 LF
(Wisdom-Holman map)

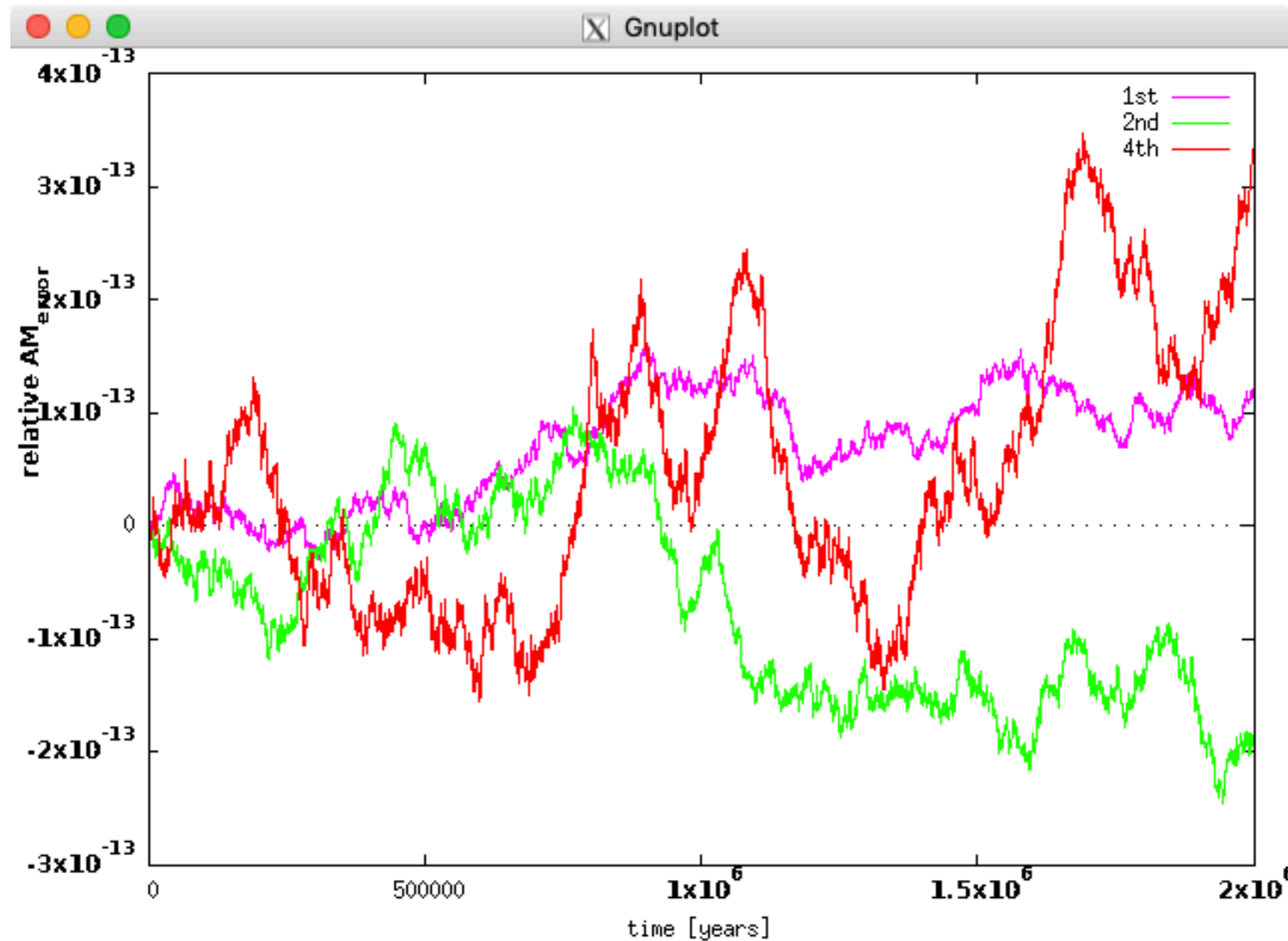
問題に応じてハミルトニアンを分割方法を変える
→ 効率良く高精度化・高速化が可能
(正準摂動論の枠組みで定式化される)

4次の方法の"うなり"は出力間隔のエイリアシング、または何らかの内部共鳴。誤差の平均値は増減しない

1.84394e+06, 2.52983e-07

N体の計算例 (N=6, 太陽+木星土天海冥)

全角運動量の
相対誤差



- 1次
- 2次 (LF)
- 4次

次数による
誤差の差異が
無い

時刻

まとめ

- 常微分方程式の数値解法には多くの種類がある
- 解くべき問題に応じて最適な方法を選択すべし
- 計算手法の前にモデルや初期値をよく検討せよ

参考にした資料 – 前半(汎用的な解法)

スライド中で引用されたものを除く

- https://www2.yukawa.kyoto-u.ac.jp/~norihito.tanahashi/pdf/ODE/note_1.pdf 常微分方程式一般
- <http://www.phys.u-ryukyu.ac.jp/%7Eemaeno/sizensuugaku2016/lec17.html> 常微分方程式一般
- https://www1.gifu-u.ac.jp/~tanaka/numerical_analysis.pdf 汎用的な数値解法
- <http://www.caero.mech.tohoku.ac.jp/publicData/Daiguji/Chapter7.pdf> 汎用的な数値解法
- <http://www.na.scitec.kobe-u.ac.jp/~yamamoto/lectures/algorithms2007/chapter2.PDF> 汎用的な数値解法
- http://www.slis.tsukuba.ac.jp/~fujisawa.makoto.fu/lecture/mic/text/08_derivative1.pdf 汎用的な数値解法
- <http://maya.phys.kyushu-u.ac.jp/~knomura/education/numerical-physics/text5/node2.html> 修正オイラー法
- <https://qiita.com/Dason08/items/d4c15283359d1bd83ae3> 修正オイラー法
- <https://www.inaba-lab.org/wiki/index.php/シミュレーション> 打ち切り誤差
- https://www.cfca.nao.ac.jp/~cfca/hpc/muv/text/makino_07.pdf 過去のN体学校資料
- https://en.wikipedia.org/wiki/Runge%E2%80%93Kutta_methods Runge-Kutta法

参考にした資料 – 後半(正準形式の解法)

スライド中で引用されたものを除く

- [神戸大学陰山聡先生の講義資料その一](#)
- [神戸大学陰山聡先生の講義資料その二](#)
- <http://www.qi.mp.es.osaka-u.ac.jp/personal/imoto/index-j/jugyou/Kaiseki170727.pdf>
- <https://www.ms.u-tokyo.ac.jp/~asuke/kougi/kyoto/dual.pdf>
- <http://hooktail.sub.jp/differentialforms/ExteriorProds/>
- <https://qiita.com/kaityo256/items/fd5a6ff3fcf0eb7bc860>
- <https://qiita.com/kaityo256/items/a4968a8de7ed636a2e79>
- <https://academic.oup.com/pasj/article/64/2/35/1481871>
- https://www.researchgate.net/publication/200175533_Geometric_integrators_for_ODEs
- https://www.jstage.jst.go.jp/article/sicejl1962/40/6/40_6_411/_article/-char/ja/
- Yoshida (1990, preprint) → 電子版が存在しない。関心ある方は学校主催者へご連絡を

正準変換

シンプレクティック形式

シンプレクティック変換

微分形式、外積

外積(ウェッジ積)

Velocity Verlet

時間反転対称性

誤差解析の例

幾何学的積分法

吉田春夫先生による記事

全角運動量の厳密保存

参考になる資料 – 講義後の追加

汎用的な解法に関する定番の教科書

- <https://www.springer.com/gp/book/9783540566700>
- <https://www.springer.com/gp/book/9783540604525>

天体力学における多段法(対称型)の利用例

- <https://ui.adsabs.harvard.edu/abs/1999esra.conf..309F/abstract>
- <https://ui.adsabs.harvard.edu/abs/1996EM%26P..72..165K/abstract>

太陽輻射圧がある系でのシンプレクティック数値解法の利用例

- <https://ui.adsabs.harvard.edu/abs/2012AdSpR..49.1472H/abstract>

プラズマ・核融合学会誌上の粒子運動論の特集記事群

- http://www.jspf.or.jp/Journal/PDF_JSPF/jspf2015_01/jspf2015_01-jp.pdf
- http://www.jspf.or.jp/Journal/PDF_JSPF/jspf2015_02/jspf2015_02-jp.pdf
- http://www.jspf.or.jp/Journal/PDF_JSPF/jspf2015_03/jspf2015_03-jp.pdf

ご静聴ありがとうございます



Center for Computational Astrophysics