情報数学C

Mathematics for Informatics C

第8回 常微分方程式の数値解法
(オイラー法,ルンゲ･クッタ法,
予測子修正子法,陰的解法)

情報メディア創成学類
藤澤誠

今回の講義内容

- **今日の問題**
- オイラー法, ホイン法, ルンゲ･クッタ法
- アダムス･バッシュホース法, 予測子修正子法
- 後退オイラー法
- 連立常微分方程式, 高階常微分方程式

今回の講義で解く問題

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

今回の講義で解く問題

常微分方程式(ODE : Ordinary Differential Equation)

yの1次微分が関数$f(x,y)$となるとき,
その解曲線$y=g(x)$を求めよ.

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

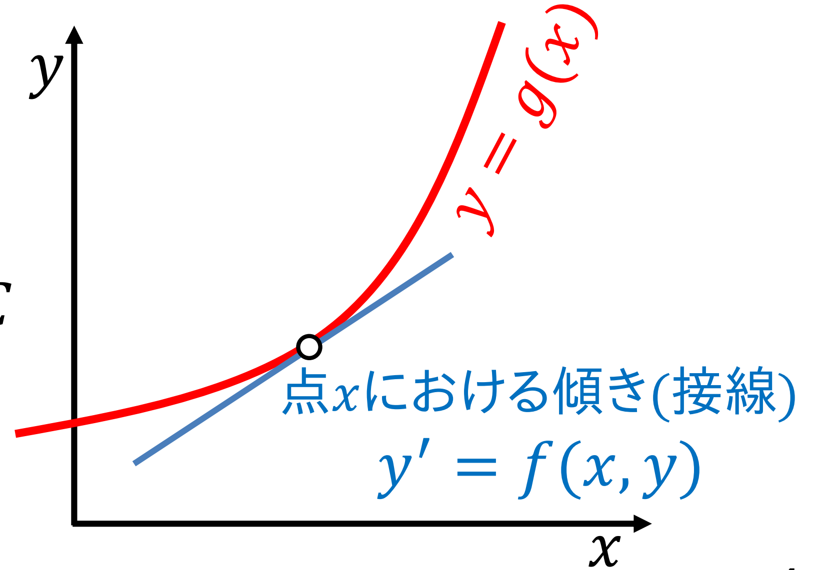
例) $\dfrac{dy}{dx} = y$

両辺にdx/yを掛けて積分

$$\int \frac{1}{y} dy = \int 1 dx \Rightarrow \log_e y = x + C$$

$\exp(C) \to C$とすると

$$y = C\exp(x)$$



今回の講義で扱う問題

常微分方程式は**どんなところで使われる?**

自然現象, 工学モデルの多くはその**支配方程式が常微分方程式/偏微分方程式*の形**になる

物理	ニュートンの運動方程式($m\frac{\partial^2 x}{\partial t^2} = F$)が微分方程式であり, 多くの力学の支配方程式はこれから導かれているため微分方程式になる
工学	熱力学の熱拡散方程式や流体力学のナビエ･ストークス方程式, 電磁気学のマクスウェル方程式, 量子力学のシュレディンガー方程式などほとんどの工学分野は微分方程式で表された支配方程式を基礎としている

これらの微分方程式のほとんどが**非線形微分方程式で数学的に解くことが難しい**(境界条件が簡単なら解けたりするけど実際の問題には適用しづらい)

*常微分方程式と偏微分方程式の違いは独立変数の数(要は$\frac{df}{dx}$か$\frac{\partial f}{\partial x} + \frac{\partial f}{\partial y}$の違い)

今回の講義で解く問題

常微分方程式を解いた結果の$y = g(x)$は**関数**?

数値計算では具体的な数値が求まる. 関数$g(x)$の形(多項式とか)が決まっていないのにどうやって何を求めるのか?

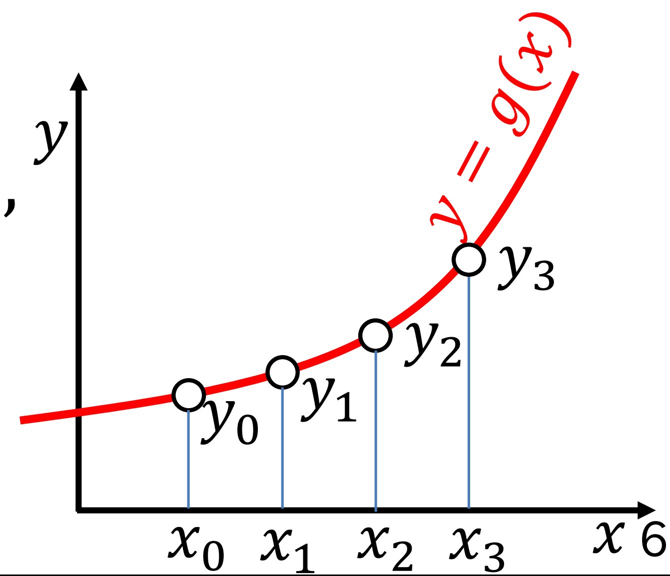
離散点$x_1, x_2, \ldots, x_n$における値$y_1, y_2, \ldots, y_n$を

- **初期値**(x_0, y_0)から順番に求めていく

⇨ **初期値問題**

- **境界条件**$[a, b]$ $(g(a) = \alpha, g(b) = \beta)$に基づいて求める

⇨ **境界値問題**



今回の講義で解く問題

今回の講義で扱う「微分方程式」についての前提条件

- 微分方程式の右辺項$f(x, y)$が**数式として与えられ,**
  初期値(x_0, y_0)もしくは境界条件が**値として得られた**とき,
  離散点x_i**における**y_i**の値**を求める$(i = 1,2, \dots)$
- 今回の講義ではyは**一つの変数**x**だけで表される**とする
  (つまり, dy/dx). $\partial y/\partial x$で表される2変数以上の場合は次回の講義で扱う(偏微分方程式:PDE).
  また今回扱うのは**初期値問題**
- $f(x, y)$は**線形でも非線形でもどのような形**でも良い
  (ただし微分は含まない). 左辺が**2階以上の導関数**となる場合もOK
- この授業で扱うのは**数値微分**であり[自動微分]ではないので注意

今回の講義内容

- 今日の問題
- **オイラー法, ホイン法, ルンゲ･クッタ法**
- アダムス･バッシュホース法, 予測子修正子法
- 後退オイラー法
- 連立常微分方程式, 高階常微分方程式

初期値問題

初期値問題について考えてみよう.

$$\frac{dy}{dx} = f(x, y) \qquad \text{:常微分方程式}$$

$$y_0 = g(x_0) \qquad \text{:初期値}(x_0, y_0)$$

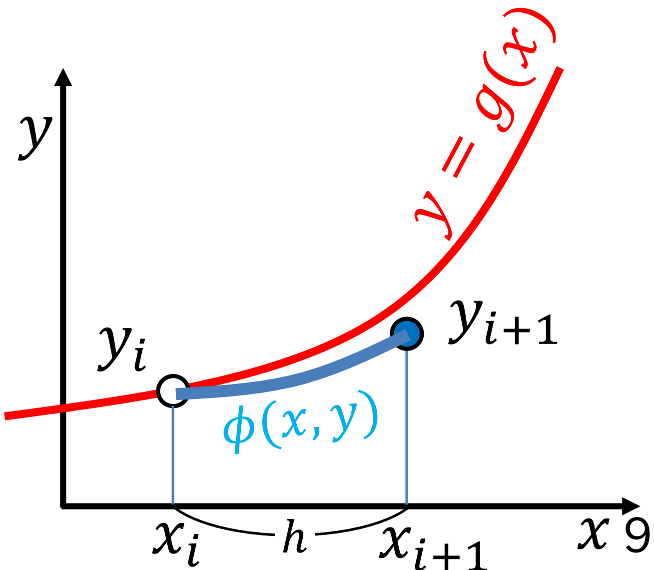
x_0でy_0になることは分かっているので, 求めたいのはそれ以降の離散点$(x_1, x_2, \ldots)$での値:$y_1, y_2, \ldots$

⇒刻み幅をhとし$(x_{i+1} = x_i + h)$,

$\boldsymbol{x_i, y_i, h}$**で**$\boldsymbol{y_{i+1}}$**が近似できる**とすると :

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

1段階法



オイラー法

1段階法の$\phi(x_i, y_i)$を具体的に考えてみよう!

解曲線$y = g(x)$で，$y_{i+1} = g(x_{i+1}) = g(x_i + h)$なので，
x_i周りで**テイラー展開**

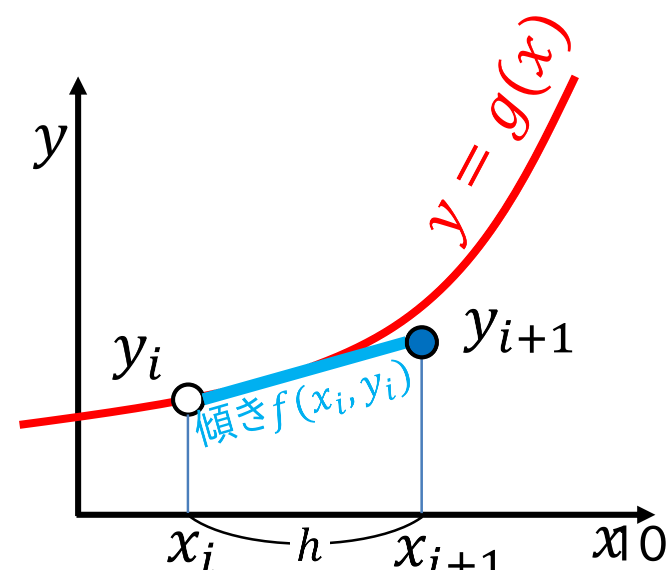
$$y_{i+1} = g(x_i) + hg'(x_i) + \frac{h^2}{2} g''(x_i) + \cdots$$

$g(x_i) = y_i, g'(x_i) = y_i' = f(x_i, y_i)$なので：

$$y_{i+1} = y_i + hf(x_i, y_i) \ + O(h^2)$$

⇩ h^2以降の項を無視

$$y_{i+1} = y_i + hf(x_i, y_i)$$



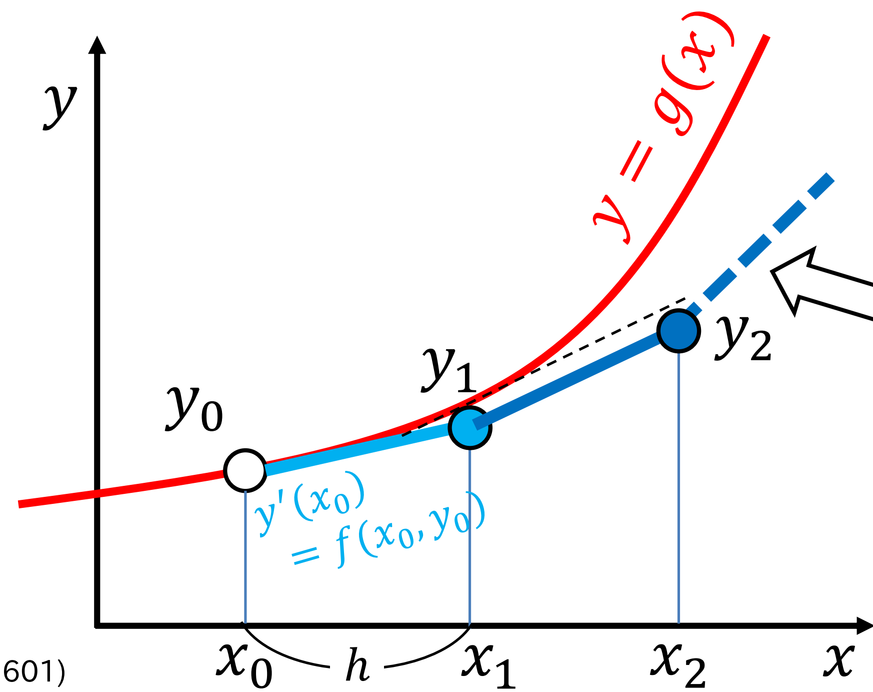
オイラー法

オイラー法

オイラー法における$\phi(x_i, y_i) = f(x_i, y_i)$は**解曲線の$x_i$における接線の傾き**

⇒ 接線の方向に解を進めていくのがオイラー法

$$y_{i+1} = y_i + hf(x_i, y_i) \qquad (i = 0,1, \dots)$$



$y_0 \to y_1 \to y_2 \to \cdots$
と$g(x)$を**近似する解**を次々と求めていく

オイラー法

オイラー法の別の考え方

オイラー法は有限範囲での微分(差分)と考えることができる

$$y_{i+1} = y_i + hf(x_i, y_i)$$

⇩ 右辺が$f(x, y)$だけになるように式変形

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

左辺について$h \to 0$の極限をとると微分の定義そのもの

$$\lim_{h \to 0} \frac{y(x_i + h) - y(x_i)}{h} = \frac{dy}{dx}$$

つまり, オイラー法は有限の大きさのhを用いてdy/dxを近似(これを**差分**という)して, 常微分方程式を解く方法*

*この考え方は次回の講義でもまた出てくるので覚えておこう

オイラー法

オイラー法の誤差

⇒ テイラー展開より:

$$y_{i+1} = \underbrace{y_i + hf(x_i, y_i)}_{\text{オイラー法}} \; \underbrace{+ \frac{h^2}{2} f'(x_i, y_i) + \cdots}_{\text{誤差}(O(h^2))}$$

1回反復で誤差が$\frac{h^2}{2}f'(x,y)$なので, n回反復後に求まるy_nの誤差は:

$$E_n \approx n\frac{h^2}{2}f'(x,y) = \frac{x_n - x_0}{2}hf'(x,y)$$

オイラー法の誤差は $O(h)$

⇒ 精度は低く, 更に計算方法から**誤差が蓄積されやすい**

ただし, $h \to 0$で真値に収束する([オイラー法の収束性](#))

オイラー法

オイラー法の計算手順

1. 初期値(x_0, y_0), 刻み幅hを設定
2. 以下の手順を繰り返す$(i = 0,1, \dots, n-1)$
   a. (x_i, y_i)での関数値$f_i = f(x_i, y_i)$の計算
   b. 次のステップのyを計算：$y_{i+1} = y_i + hf_i$
   c. xの値の更新：$x_{i+1} = x_i + h$

反復法ではないので収束条件はなく，
範囲$[x_0, x_n]$でy_iの値を計算するだけ

オイラー法

オイラー法のコード例

微分方程式を範囲$[a,b]$でn分割して解く

```
double h = (b-a)/n; // 刻み幅
double x = a;       // xの初期値
double y = y0;      // yの初期値
for(int i = 0; i <= n-1; ++i){
    double fi = func(x, y);
    y = y+h*fi; // yの更新
    x = x+h;    // xの更新
}
```

途中の値が必要ないならば, x, yの変数は1つずつでOK. 途中の値も必要なら配列に格納すること

オイラー法

オイラー法の実行結果例

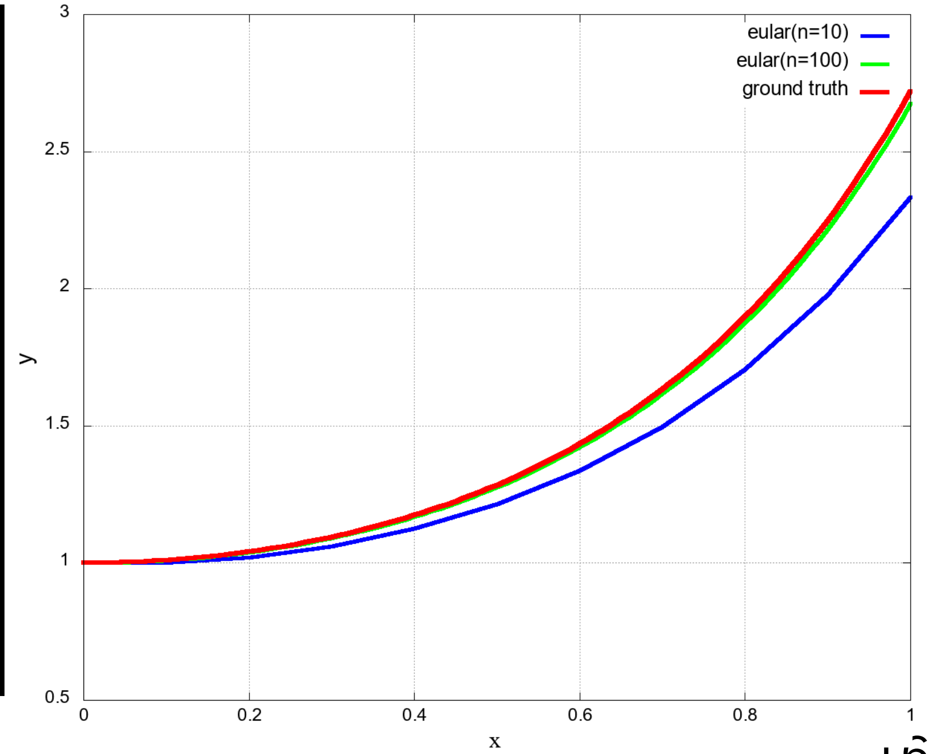
$$\frac{dy}{dx} = 2xy \quad \Rightarrow \quad f(x,y) = 2xy \qquad \text{解 } y = Ce^{x^2} = y_0 e^{x^2}$$

範囲$[0,1]$, 初期値$y_0 = 1$, 刻み幅$h = 0.1$

```
y(0.1) = 1
y(0.2) = 1.02
y(0.3) = 1.0608
y(0.4) = 1.124448
y(0.5) = 1.21440384
y(0.6) = 1.335844224
y(0.7) = 1.496145531
y(0.8) = 1.705605905
y(0.9) = 1.97850285
y(1) = 2.334633363

error = 0.3836484654
ground truth = 2.718281828
```

→ $y(1)$の誤差

→ $y(1)$の真値



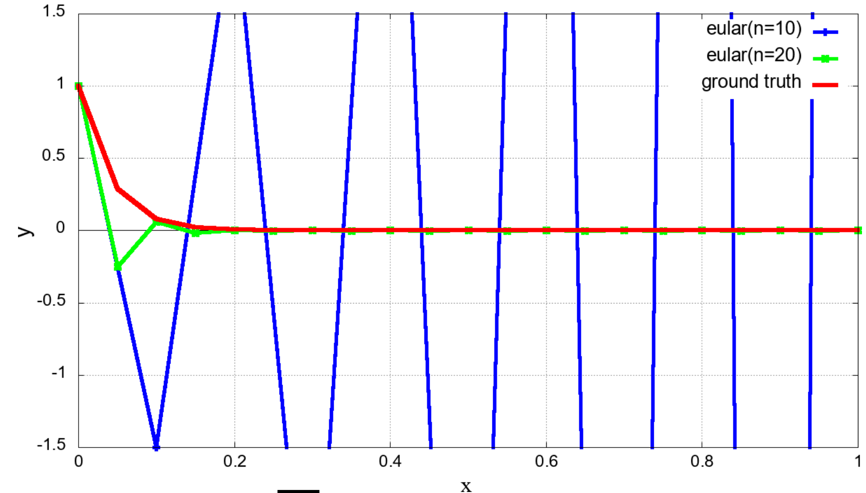
オイラー法

オイラー法の実行結果例(安定しない例*)

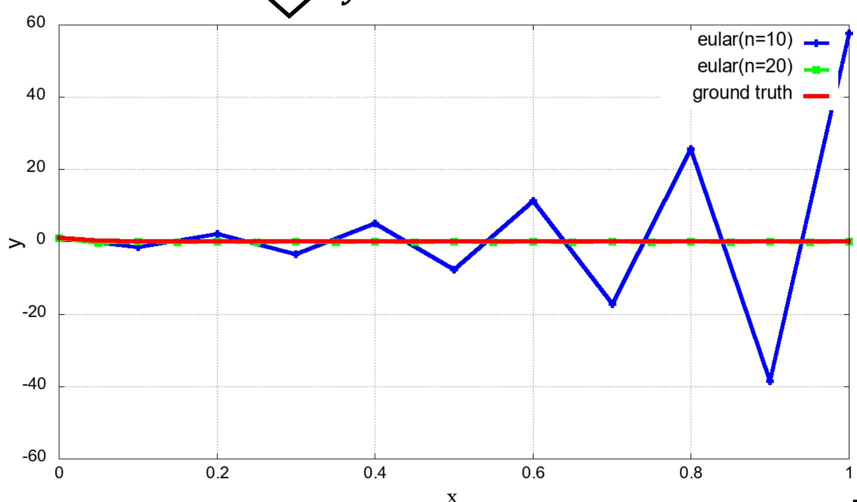
$$\frac{dy}{dx} = -25y \quad 解\ y = y_0 e^{-25x}$$

範囲$[0, 1]$, 初期値$y_0 = 1$, 刻み幅$h = 0.1$

```
y(0.1) = -1.5
y(0.2) = 2.25
y(0.3) = -3.375
y(0.4) = 5.0625
y(0.5) = -7.59375
y(0.6) = 11.390625
y(0.7) = -17.0859375
y(0.8) = 25.62890625
y(0.9) = -38.44335938
y(1) = 57.66503906

error = 57.66503906  → y(1)の誤差
ground truth = 1.3887943e-11  → y(1)の真値
```



y軸のスケールを大きく



刻み幅を小さくしないと安定しないものを**硬い方程式**と呼ぶ

*$y' = \lambda y$の形の方程式は単純だけど色々な現象をモデル化するのに用いられている

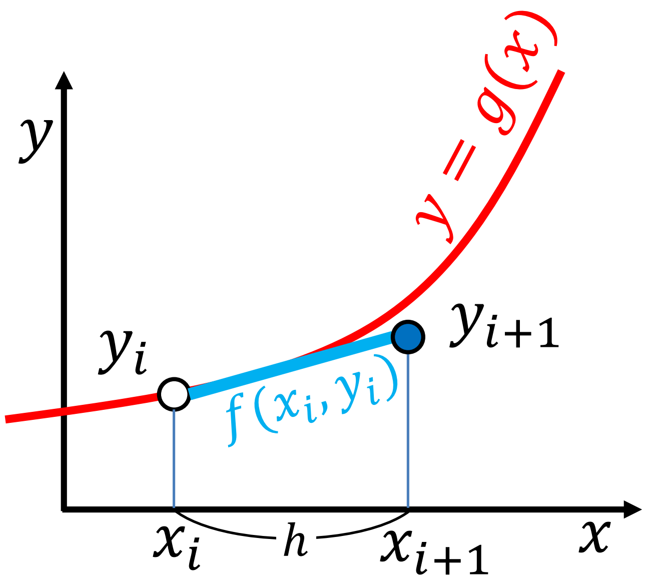
ホイン法

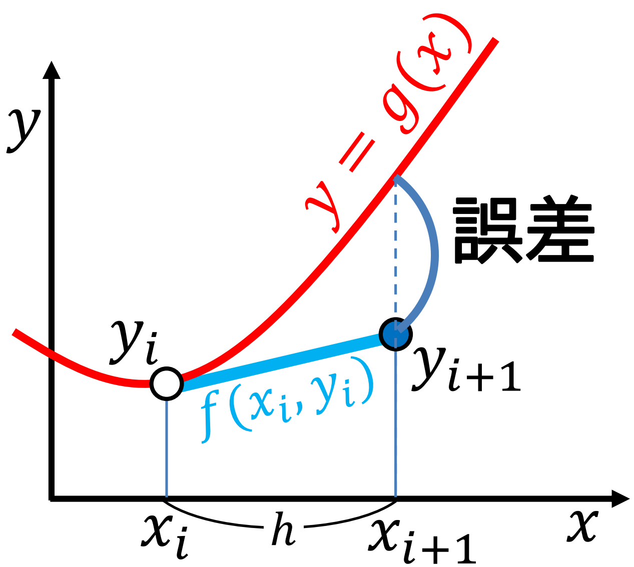
$O(h)$の誤差はさすがに**使いづらい**(収束性から検証には使えるけど)

⇒ もっと誤差を小さくするためには？

オイラー法は**区間の始点x_iでの傾き$y'_i = f(x_i, y_i)$のみ**を使っているので**y'の変化が大きいときに誤差が大きい**

⇒始点だけでなく終点x_{i+1}での傾きも使えないか？





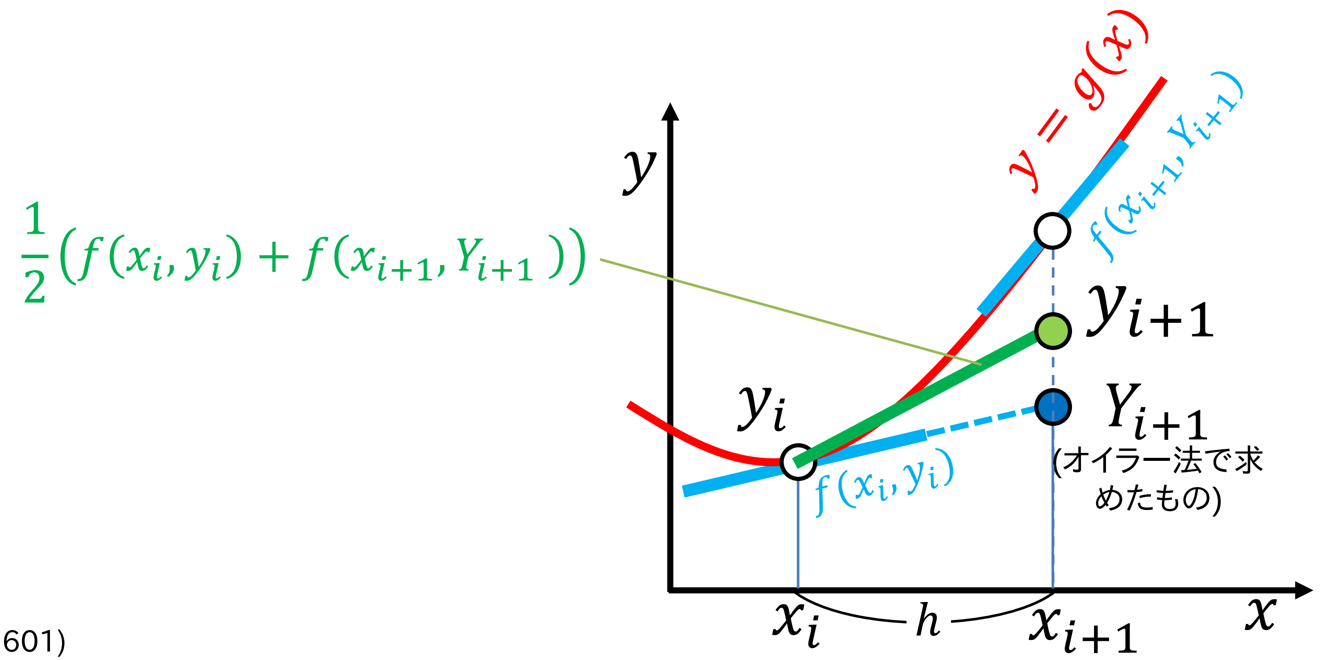
ホイン法

区間始点x_iでの傾き$f(x_i, y_i)$と終点x_{i+1}での傾き$f(x_{i+1}, y_{i+1})$の**平均**を用いて更新

$$y_{i+1} = y_i + \frac{h}{2}\left(f(x_i, y_i) + f(x_{i+1}, Y_{i+1})\right)$$

ホイン法

ただし, Y_{i+1}はオイラー法で求めた値$(Y_{i+1} = y_i + hf(x_i, y_i))$



ホイン法

ホイン法の誤差

ホイン法の式はテイラー展開からも求められるのでそこから誤差を算出してみよう.

オイラー法でx_iから$x_{i+1} = x_i + h$での$y_{i+1} = y_i + hf(x_i, y_i)$を求めたとき, $f(x_{i+1}, y_{i+1})$は2変数のテイラー展開より:

$$f(x_{i+1}, y_{i+1}) = f(x_i + h, y_i + hf(x_i, y_i))$$
$$= f(x_i, y_i) + h\left(\underline{\frac{\partial f(x_i, y_i)}{\partial x} + f(x_i, y_i)\frac{\partial f(x_i, y_i)}{\partial y}}\right) + \cdots$$

また, yの2次微分$y'' = f'$は全微分($\frac{df}{dx} = \frac{\partial f}{\partial x}\frac{dx}{dx} + \frac{\partial f}{\partial y}\frac{dy}{dx}$)を使うと

$$y'' = \frac{df\big(x, y(x)\big)}{dx} = \underline{\frac{\partial f\big(x, y(x)\big)}{\partial x} + \frac{\partial f\big(x, y(x)\big)}{\partial y}y'}$$

$y' = f(x, y)$
なので同じ

2つの式から：

$$\boxed{y'' = \frac{f\big(x_{i+1}, y_i + hf(x_i, y_i)\big) - f(x_i, y_i) - O(h^2)}{h}}$$

ホイン法

ホイン法の誤差

前ページ青枠の式を$y_{i+1} = g(x_i + h)$のテイラー展開の式に代入:

$$
\begin{aligned}
y_{i+1} &= y_i + hy'(x_i) + \frac{h^2}{2} y''(x_i) + \cdots \\
&= y_i + hf(x_i, y_i) + \frac{h^2}{2} \frac{f\left(x_{i+1}, y_i + hf(x_i, y_i)\right) - f(x_i, y_i) - O\left(h^2\right)}{h} + O(h^3) \\
&= \underbrace{y_i + \frac{h}{2}\left(f(x_i, y_i) + f\left(x_{i+1}, y_i + hf(x_i, y_i)\right)\right)}_{\text{ホイン法の式}} + O(h^3)
\end{aligned}
$$

1回反復での誤差は$O(h^3)$
n回反復後に求まるy_nの誤差は$O(h^2)$

ホイン法

数値微分と数値積分の関係

ホイン法の式は前回やった**台形公式に似ている?**

$y' = f(x, y)$をx_iからx_{i+1}まで積分してみると:

$$\int_{x_i}^{x_{i+1}} y'(x)dx = \int_{x_i}^{x_{i+1}} f(x, y)dx$$

$y(x_{i+1}) - y(x_i)$
$= y_{i+1} - y_i$となるので

$$y_{i+1} = y_i + \int_{x_i}^{x_{i+1}} f(x, y)dx$$

この部分を**区分求積法(左端)**で近似すると**オイラー法**,
台形公式で近似するのが**ホイン法**

$\Rightarrow$ **中点法**を使えばオイラー法でも$O(h^2)$になる?

修正オイラー法

中点を用いた微分の近似

数値積分の中点法を前ページの式に代入

数値積分の中点法

$$I_i \approx h\,f\left(\frac{x_i + x_{i+1}}{2}\right)$$

$$y_{i+1} = y_i + h\,f\left(\frac{x_i + x_{i+1}}{2}, y\left(\frac{x_i + x_{i+1}}{2}\right)\right)$$

修正オイラー法

$y\left(\frac{x_i+x_{i+1}}{2}\right)$は刻み幅を$\frac{h}{2}$にしたオイラー法で計算:

$$y\left(\frac{x_i + x_{i+1}}{2}\right) = y_i + \frac{h}{2} f(x_i, y_i)$$

精度は中点法を用いているので$O(h^2)$

ホイン法

ホイン法のコード例

微分方程式を範囲$[a, b]$でn分割して解く

```
double h = (b-a)/n; // 刻み幅
double x = a;  // xの初期値
double y = y0; // yの初期値
for(int i = 0; i <= n-1; ++i){
    // Y(i+1) をオイラー法で求める
    double fi = func(x, y);
    double Yi = y+h*fi; // Yiの計算

    // f(x,y)とY(i+1)の平均でyを更新
    y = y+h*(fi+func(x+h, Yi))/2;
    x = x+h;    // xの更新
}
```

$Y_{i+1} = y_i + hf(x_i, y_i)$：オイラー法で$x_{i+1} = x_i + h$での近似解を求める

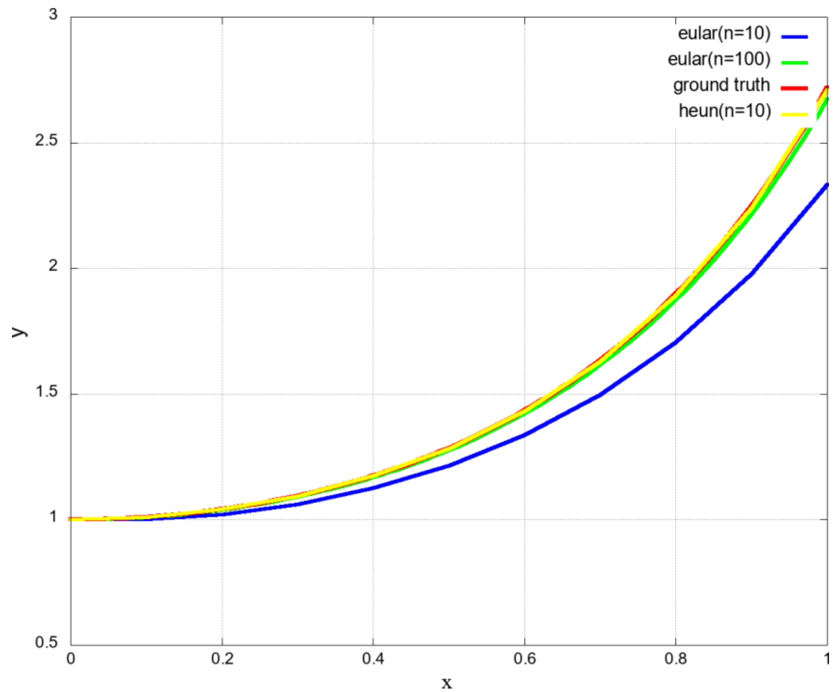
$y_{i+1} = y_i + \frac{h}{2}\left(f(x_i, y_i) + f(x_{i+1}, Y_{i+1})\right)$

ホイン法

ホイン法の実行結果例　$\frac{dy}{dx} = 2xy$　解 $y = Ce^{x^2} = y_0 e^{x^2}$

範囲$[0, 1]$, 初期値$y_0 = 1$, 刻み幅$h = 0.1$

```
y(0.1) = 1.01
y(0.2) = 1.040704
y(0.3) = 1.093988045
y(0.4) = 1.173192779
y(0.5) = 1.2834729
y(0.6) = 1.432355757
y(0.7) = 1.630593794
y(0.8) = 1.893445513
y(0.9) = 2.242596866
y(1) = 2.709057014

error = 0.009224814449 → y(1)の誤差
ground truth = 2.718281828→ y(1)の真値
```



10分割のオイラー法 (誤差$E \approx 0.38365$)や100分割のオイラー法 (誤差$E \approx 0.04449$)よりも精度は高い

ホイン法

ホイン法の実行結果例(硬い方程式)

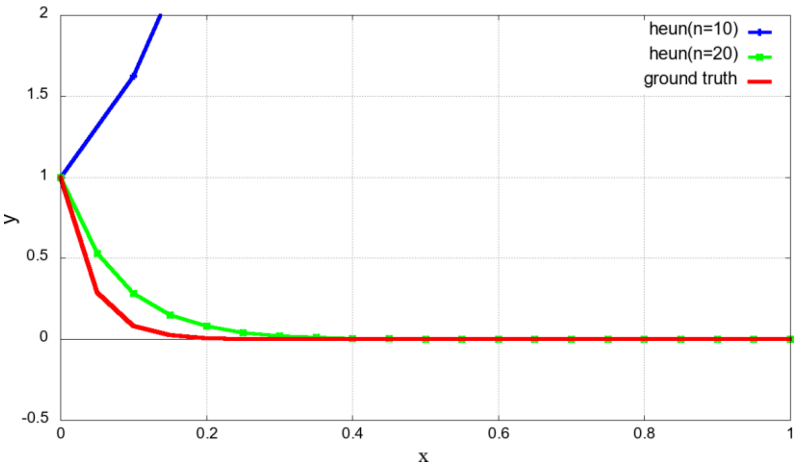
$$\frac{dy}{dx} = -25y \quad 解\ y = y_0 e^{-25x}$$

範囲$[0, 1]$, 初期値$y_0 = 1$, 刻み幅$h = 0.1$

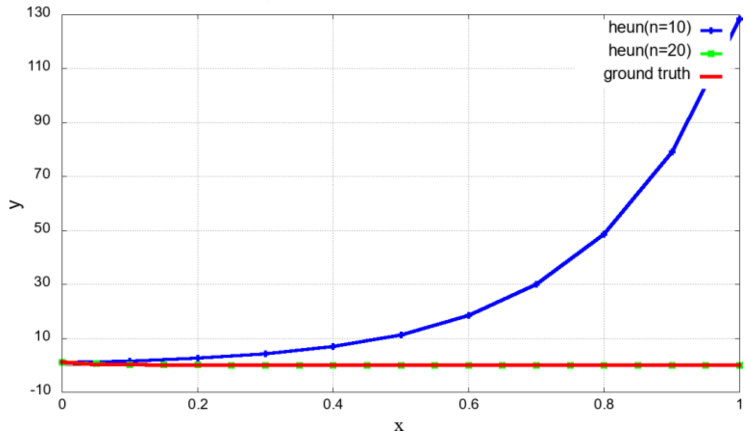
```
y(0.1) = 1.625
y(0.2) = 2.640625
y(0.3) = 4.291015625
y(0.4) = 6.972900391
y(0.5) = 11.33096313
y(0.6) = 18.41281509
y(0.7) = 29.92082453
y(0.8) = 48.62133986
y(0.9) = 79.00967727
y(1) = 128.3907256

error = 128.3907256
ground truth = 1.3887943e-11
```

→ $y(1)$の誤差

→ $y(1)$の真値



⇩ y軸のスケールを大きく



硬い問題については振動はしなくなったが刻み幅$h = 0.1$だと解が発散している($h = 0.05$はオイラー法より安定)

ルンゲ･クッタ法

ホイン法では2点での$hf(x,y)$の**平均をとることで精度が向上**した

⇒ 3点, 4点, ... と増やしていけば**近似精度をより向上**させることができるのでは?

$$y_{i+1} = y_i + h(b_1 k_1 + b_2 k_2 + \cdots + b_m k_m)$$

ここで

$$k_1 = f(x_i, y_i) \qquad \Rightarrow c_1 = 0, a_{11} = 0$$
$$k_2 = f(x_i + c_2 h, y_i + h a_{21} k_1)$$
$$k_3 = f(x_i + c_3 h, y_i + h(a_{31} k_1 + a_{32} k_2))$$
$$k_4 = f(x_i + c_4 h, y_i + h(a_{41} k_1 + a_{42} k_2 + a_{43} k_3))$$
$$\vdots$$

この式による近似を **ルンゲ･クッタ法** という

- k_mまで使ってl次精度の場合 ***m*段*l*次のルンゲ･クッタ公式**と呼ぶ

ルンゲ・クッタ法

1段1次のルンゲ・クッタ法(RK1) (1次精度:1区間の誤差は$O(h^2)$)

$$y_{i+1} = y_i + hb_1k_1 \qquad k_1 = f(x_i, y_i)$$

$b_1 = 1$とするとオイラー法の式と同じ

⇒ オイラー法 = 1次のルンゲ・クッタ法

2段2次のルンゲ・クッタ法(RK2) (2次精度:1区間の誤差は$O(h^3)$)

$$y_{i+1} = y_i + h(b_1k_1 + b_2k_2)$$

$$k_1 = f(x_i, y_i)$$
$$k_2 = f(x_i + c_2h, y_i + ha_{21}f(x_i, y_i))$$

$c_2 = a_{21} = 1, b_1 = b_2 = 1/2$とするとホイン法

$c_2 = a_{21} = 1/2, b_1 = 0, b_2 = 1/2$とすると修正オイラー法

⇒ ホイン法,修正オイラー法 = 2次のルンゲ・クッタ法

ルンゲ・クッタ法

3段3次のルンゲ・クッタ法(RK3) (3次精度: 1区間の誤差は$O(h^4)$)

$$y_{i+1} = y_i + h(b_1 k_1 + b_2 k_2 + b_3 k_3)$$

$$k_1 = f(x_i, y_i)$$
$$k_2 = f(x_i + c_2 h, y_i + h a_{21} k_1)$$
$$k_3 = f(x_i + c_3 h, y_i + h a_{31} k_1 + h a_{32} k_2)$$

係数をどうやって決めるのか?

⇒ **3次精度**を得るためには**テイラー展開の3次項**まで近似する必要がある

$$y_{i+1} = y(x_i + h) = y(x_i) + h\left(y'(x_i) + \frac{h}{2} y''(x_i) + \frac{h^2}{6} y'''(x_i)\right) + O(h^4)$$

ここで $y'(x_i) = f(x_i, y(x_i))$ であり, また, y'', y'''は全微分を用いると*:

$$y''(x) = f'(x,y) = \frac{\partial f}{\partial x} + \frac{\partial f}{\partial y} f = f_x + f_y f$$

$$y'''(x_n) = \frac{\partial f'}{\partial x} + \frac{\partial f'}{\partial y} f = f_{xx} + 2 f_{xy} f + f_{yy} f^2 + f_y f_x + f_y^2 f$$

*スペースの関係上$f(x,y) = f, \frac{\partial f}{\partial x} = f_x, \frac{\partial^2 f}{\partial x^2} = f_{xx}$といった表記をこれ以降使います

ルンゲ・クッタ法

k_2, k_3のテイラー展開(2次項まで):

テイラー展開後, $k_1 = f$を代入

$$k_2 = f(x_i + c_2 h, y_i + h a_{21} k_1)$$
$$= f + \left(c_2 f_x + a_{21} f_y f\right)h + \left(\frac{c_2}{2} f_{xx} + c_2 a_{21} f_{xy} f + \frac{a_{21}^2}{2} f_{yy} f^2\right)h^2 + O(h^3)$$

$$k_3 = f(x_i + c_3 h, y_i + h a_{31} k_1 + h a_{32} k_2)$$
$$= f + \left(c_3 f_x + (a_{31} k_1 + a_{32} k_2) f_y\right)h$$
$$+ \left(\frac{c_3}{2} f_{xx} + c_3 (a_{31} k_1 + a_{32} k_2) f_{xy} + \frac{(a_{31} k_1 + a_{32} k_2)^2}{2} f_{yy}\right)h^2 + O(h^3)$$

$$\Downarrow$$

$k_1 = f$と上のk_2の式を代入

$$k_3 = f + \left(c_3 f_x + (a_{31} + a_{32}) f_y f\right)h$$
$$+ \left(\frac{c_3}{2} f_{xx} + c_3 (a_{31} + a_{32}) f_{xy} f + \frac{(a_{31} + a_{32})^2}{2} f_{yy} f^2 + c_2 a_{32} f_x f_y + a_{21} a_{32} f_y^2 f\right)h^2$$
$$+ O(h^3)$$

ルンゲ・クッタ法

$(b_1k_1 + b_2k_2 + b_3k_3)$に代入したものが前々ページの$y_{i+1}$の**テイラー展開と一致すれば良い**　$\Rightarrow h^0, h^1, h^2$の項に分けて整理する

$$(b_1k_1 + b_2k_2 + b_3k_3) = (b_1 + b_2 + b_3)f \qquad \Rightarrow h^0\text{の項}$$

$$+\left\{(b_2c_2 + b_3c_3)f_x + (b_2a_{21} + b_3(a_{31} + a_{32}))f_yf\right\}h \qquad \Rightarrow h^1\text{の項}$$

$$+\left\{\frac{b_2c_2^2 + b_3c_3^2}{2}f_{xx} + \left(b_2c_2a_{21} + b_3c_3(a_{31} + a_{32})\right)f_{xy}f + \frac{b_2a_{21}^2 + b_3(a_{31} + a_{32})^2}{2}f_{yy}f^2 + b_3c_2a_{32}f_xf_y + b_3a_{21}a_{32}f_y^2f\right\}h^2 \qquad \Rightarrow h^2\text{の項}$$

y_{i+1}のテイラー展開式にy', y'', y'''の式を代入したもの

$$y_{i+1} \approx y_i + h\left(1 \cdot f + \left(\frac{1}{2}f_x + \frac{1}{2}f_yf\right)h + \left(\frac{1}{6}f_{xx} + \frac{1}{3}f_{xy}f + \frac{1}{6}f_{yy}f^2 + \frac{1}{6}f_xf_y + \frac{1}{6}f_y^2f\right)h^2\right)$$

2つの式でf_xやf_yfなどの項を比較するとb_i, c_i, a_{ij}に関する式が複数作れる

ルンゲ・クッタ法

b_i, c_i, a_{ij}に関する式:

$$
\begin{cases}
b_1 + b_2 + b_3 = 1 \\
b_2 c_2 + b_3 c_3 = \dfrac{1}{2} \\
b_2 a_{21} + b_3(a_{31} + a_{32}) = \dfrac{1}{2} \\
b_2 c_2^2 + b_3 c_3^2 = \dfrac{1}{3} \\
b_2 c_2 a_{21} + b_3 c_3 (a_{31} + a_{32}) = \dfrac{1}{3} \\
b_2 a_{21}^2 + b_3 (a_{31} + a_{32})^2 = \dfrac{1}{6} \\
b_3 c_2 a_{32} = \dfrac{1}{6} \\
b_3 a_{21} a_{32} = \dfrac{1}{6}
\end{cases}
$$

未知数8個
$(b_1, b_2, b_3, c_2, c_3, a_{21}, a_{31}, a_{32})$
に対して式が8個なので解ける?
⇒ 実際には意味のある式は6個に
なってしまうので解は無限に存在

⇩

式を満たす解を仮定:
例) クッタの3次公式

$$b_1 = \frac{1}{6}, b_2 = \frac{2}{3}, b_3 = \frac{1}{6},$$
$$c_2 = \frac{1}{2}, c_3 = 1,$$
$$a_{21} = \frac{1}{2}, a_{31} = -1, a_{32} = 2$$

ルンゲ・クッタ法

3段3次のルンゲ・クッタ法(RK3) (3次精度:1区間の誤差は$O(h^4)$)

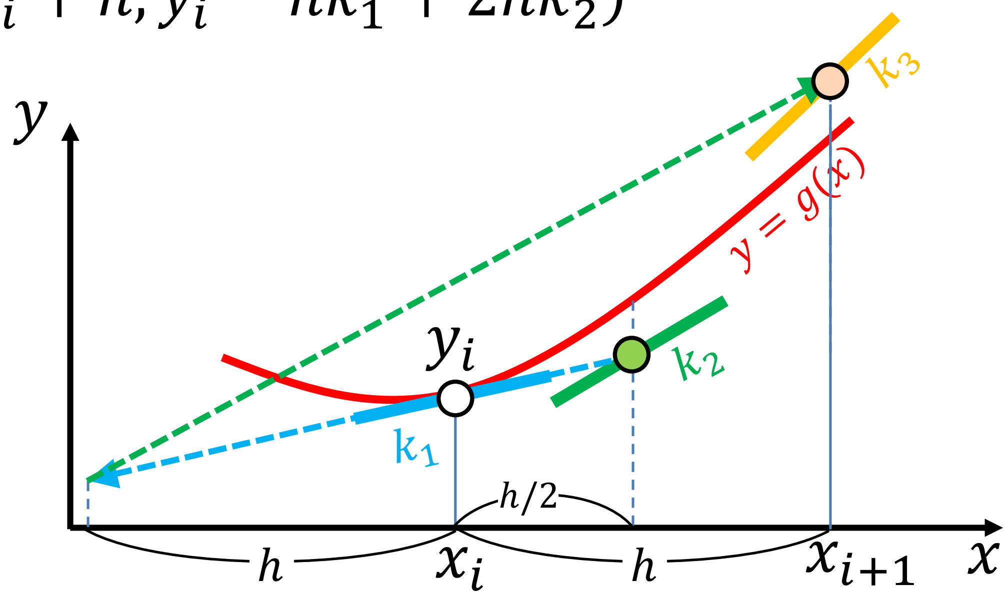
クッタの3次公式を用いて導出した式

$$y_{i+1} = y_i + \frac{h}{6}(k_1 + 4k_2 + k_3)$$

$$k_1 = f(x_i, y_i)$$

$$k_2 = f\left(x_i + \frac{h}{2}, y_i + \frac{h}{2}k_1\right)$$

$$k_3 = f(x_i + h, y_i - hk_1 + 2hk_2)$$



ルンゲ・クッタ法

4段4次のルンゲ・クッタ法(RK4) (4次精度:1区間の誤差は$O(h^4)$)

3段3次と同様にして導出可能*

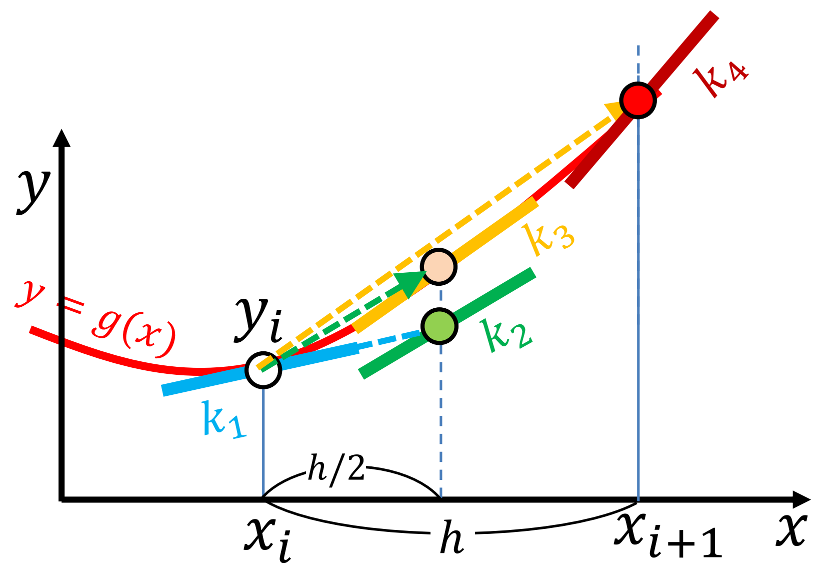
$$y_{i+1} = y_i + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

$$k_1 = f(x_i, y_i)$$

$$k_2 = f\left(x_i + \frac{h}{2}, y_i + \frac{h}{2}k_1\right)$$

$$k_3 = f\left(x_i + \frac{h}{2}, y_i + \frac{h}{2}k_2\right)$$

$$k_4 = f(x_i + h, y_i + hk_3)$$



*導出過程は長すぎるのでこの講義内では説明しませんが，興味のある人は自分で導出してみよう!

ルンゲ・クッタ法

ルンゲ・クッタ法について

- 一番よく使われるのがRK4. 今回導出したような$a_{ii}=0$となる場合は**陽的ルンゲクッタ法**と呼ぶ
- n段m次の**n段は計算コスト, m次は精度**を表す. $n=m$とできるのは4段4次まで. それ以上の場合は$n>m$となり**精度に対して計算コストが大きく**なってしまう*(これがRK4がよく使われる理由)
- より精度が必要な場合は**埋め込み型ルンゲ・クッタ法**や**陰的ルンゲ・クッタ法**を使う
  - **埋め込み型(適応型)RK**：段数次数の異なるRKの解から誤差を評価して, 刻み幅を1/2倍や2倍にしていく方法. 有名なのは4次と5次を使うルンゲ・クッタ-フェールベルク法(RKF45)
  - **陰的RK**：a_{ij}で$i\leq j$の項まで入れて, 連立方程式を解くことで係数を求める方法. 問題としては難しくなっているが$n<m$も可能. 有名なのは数値積分のガウス・ルジャンドル公式を用いるルンゲ・クッタ-ガウス・ルジャンドル法(2段4次, 3段6次など)

*6段5次や17段10次, 35段14次などがある

ルンゲ･クッタ法

ルンゲ･クッタ法のコード例(RK4)

微分方程式を範囲$[a, b]$でn分割して解く

```
double h = (b-a)/n; // 刻み幅
double x = a;  // xの初期値
double y = y0; // yの初期値
double k1, k2, k3, k4;
for(int i = 0; i <= n-1; ++i){
    k1 = func(x, y);
    k2 = func(x+h/2, y+(h/2)*k1);
    k3 = func(x+h/2, y+(h/2)*k2);
    k4 = func(x+h, y+h*k3);

    y = y+(h/6)*(k1+2*k2+2*k3+k4); // yの更新
    x = x+h;    // xの更新
}
```

$k_1 \sim k_4$を順番に計算

y_{i+1}を計算

ルンゲ･クッタ法

ルンゲ･クッタ法の実行結果例* $\frac{dy}{dx} = 2xy$ 解 $y = y_0 e^{x^2}$

範囲$[0, 1]$, 初期値$y_0 = 1$, 刻み幅$h = 0.1$で$y = 1$まで計算

```
[RK4]
y(1) = 2.718270175,  error = 1.165307551e-05

[RK3]
y(1) = 2.7183378,  error = 5.597151256e-05

[Heun method]
y(1) = 2.709057014,  error = 0.009224814449
[Eular method]
y(1) = 2.334633363,  error = 0.3836484654

ground truth = 2.718281828
```

オイラー法→ホイン法→RK3→RK4と**誤差が小さく**なっている.
ただし, 問題によってはRKでも振動して解が発散することはあるので注意

*見た目で差が分からないぐらいに誤差が小さくなっているのでグラフはなし

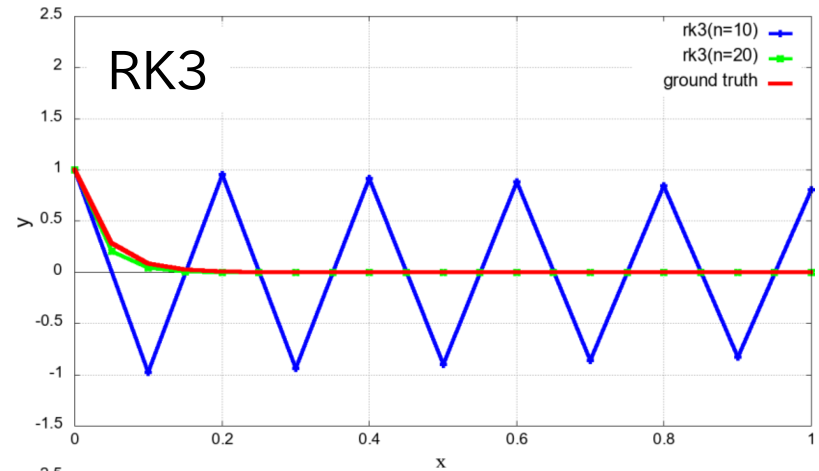
ルンゲ･クッタ法

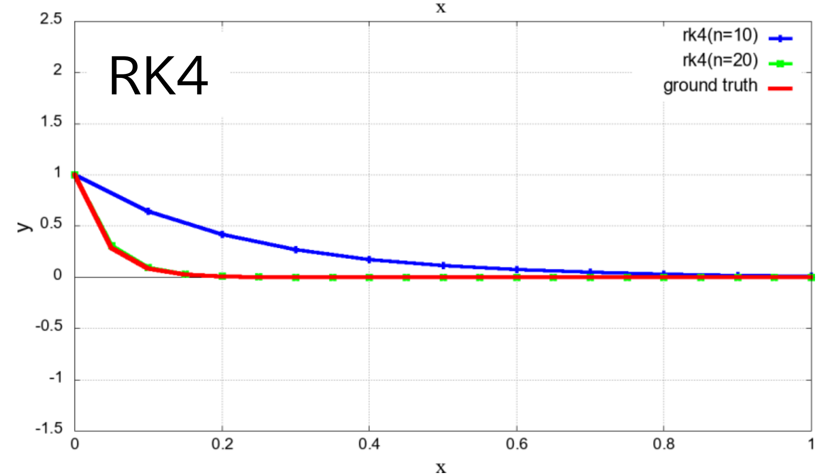
ルンゲ･クッタ法の実行結果例(硬い方程式)

$$\frac{dy}{dx} = -25y \qquad \text{解 } y = y_0 e^{-25x}$$

範囲$[0, 1]$, 初期値$y_0 = 1$, 刻み幅$h = 0.1$

```
y(0.1) = 0.6484375
y(0.2) = 0.4204711914
y(0.3) = 0.2726492882
y(0.4) = 0.1767960228
y(0.5) = 0.114641171
y(0.6) = 0.07433763434
y(0.7) = 0.04820330977
y(0.8) = 0.03125683368
y(0.9) = 0.02026810309
y(1) = 0.0131425981

error = 0.009224814449 → y(1)の誤差
ground truth = 1.3887943e-11 → y(1)の真値
```





刻み幅$h = 0.1$でRK3は発散はしていないが振動している. RK4はxが小さいときは精度が悪いが, 解は安定している

今回の講義内容

- 今日の問題
- オイラー法, ホイン法, ルンゲ・クッタ法
- **アダムス・バッシュホース法, 予測子修正子法**
- 後退オイラー法
- 連立常微分方程式, 高階常微分方程式

多段階法

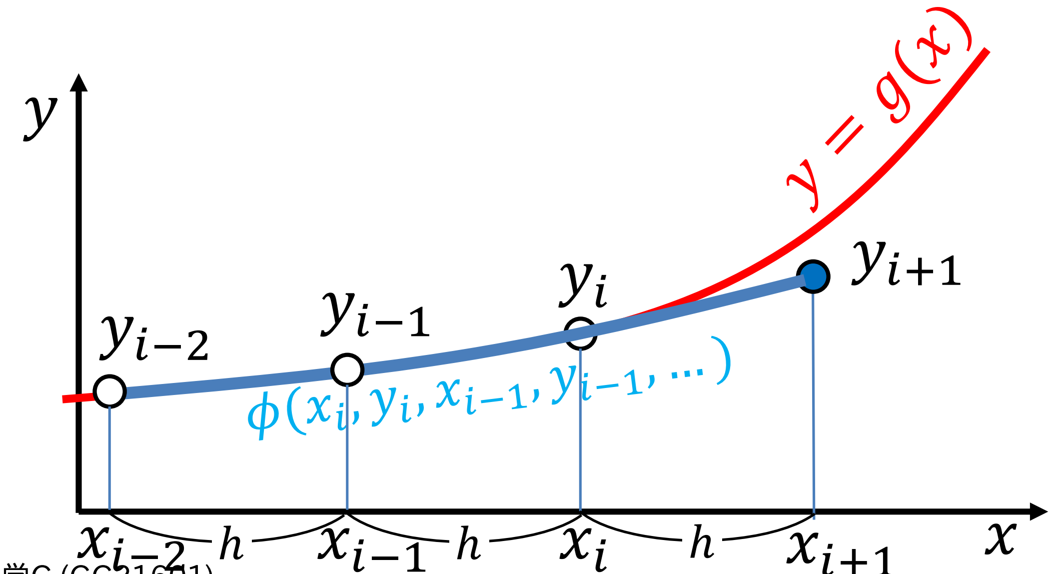
ここまで説明した1段階法では(x_i, y_i)のみから(x_{i+1}, y_{i+1})を近似していた

$\Rightarrow$ (x_{i-1}, y_{i-1})や(x_{i-2}, y_{i-2})がすでに求まっているならばそれらを使って**精度が上げられないか?**

刻み幅をhとし$x_i, y_i, x_{i-1}, y_{i-1}, \ldots$で$y_{i+1}$が近似できるとすると：

$$y_{i+1} = y_i + h\phi(x_i, y_i, x_{i-1}, y_{i-1}, x_{i-2}, y_{i-2}, \ldots)$$

多段階法



y_{i-1}やy_{i-2}は真値ではなく，そこまでの計算で求まった近似値であることに注意
(1段階法と同じく与えるのは初期値と刻み幅)

アダムス·バッシュホース法

常微分方程式は積分の形に表せることを思いだそう!

$$y_{i+1} = y_i + \int_{x_i}^{x_{i+1}} f(x,y)dx$$

積分内の$f(x,y)$を$(x_i, y_i), (x_{i-1}, y_{i-1}), (x_{i-2}, y_{i-2})$ の3点を使って**2次のラグランジュ補間多項式**で近似してみる

$$f(x,y) \approx \sum_{k=i-2}^{i} \prod_{j=i-2\sim i, j\neq k} \frac{x - x_j}{x_k - x_j} f_k$$

ここで $f_i = f(x_i, y_i)$

$$= \frac{x - x_{i-1}}{\underbrace{x_{i-2} - x_{i-1}}_{-h}} \frac{x - x_i}{\underbrace{x_{i-2} - x_i}_{-2h}} f_{i-2} + \frac{x - x_{i-2}}{\underbrace{x_{i-1} - x_{i-2}}_{h}} \frac{x - x_i}{\underbrace{x_{i-1} - x_i}_{-h}} f_{i-1} + \frac{x - x_{i-2}}{\underbrace{x_i - x_{i-2}}_{2h}} \frac{x - x_{i-1}}{\underbrace{x_i - x_{i-1}}_{h}} f_i$$

⇩

$x_i - x_{i-1} = h$よりそれぞれの分母はhで表せる

ラグランジュ補間多項式

$$L_n(x) = \sum_{i=0}^{n} f_i\, l_i(x),\ l_i(x) = \prod_{j=0\sim n, j\neq i} \frac{x - x_j}{x_i - x_j}$$

アダムス･バッシュホース法

積分の式に代入すると:

$$y_{i+1} = y_i + \frac{f_{i-2}}{2h^2}\int_{x_i}^{x_{i+1}}(x - x_{i-1})(x - x_i)dx - \frac{f_{i-1}}{h^2}\int_{x_i}^{x_{i+1}}(x - x_{i-2})(x - x_i)dx$$

$$+\frac{f_i}{2h^2}\int_{x_i}^{x_{i+1}}(x - x_{i-2})(x - x_{i-1})dx$$

⇩ それぞれ展開して積分を計算

$$y_{i+1} = y_i + h\left(\frac{5}{12}f_{i-2} - \frac{16}{12}f_{i-1} + \frac{23}{12}f_i\right)$$

2次以外にも1次や3次の場合でも近似式ができる:

$m = 1$ (2点)： $y_{i+1} = y_i + h\left(-\frac{1}{2}f_{i-1} + \frac{3}{2}f_i\right)$

$m = 3$ (4点)： $y_{i+1} = y_i + h\left(-\frac{9}{24}f_{i-3} + \frac{37}{24}f_{i-2} - \frac{59}{24}f_{i-1} + \frac{55}{24}f_i\right)$

アダムス・バッシュホース法

このようにして，m次のラグランジュ補間多項式を用いて
微分を近似する方法を **アダムス・バッシュホース法** という

アダムス・バッシュホース法の誤差

ラグランジュ補間の誤差からm次($m+1$段)のアダムス・バッシュホース法の誤差は $O(h^{m+1})$
⇒ **3点使うと3次精度，4点なら4次精度**

アダムス・バッシュホース法で計算する時の注意点

3点を用いる場合，$i \geq 2$でないとy_{i+1}を計算できない．$i = 0,1$の時はオイラー法やルンゲ・クッタ法などの1段階法を使う必要がある

アダムス・バッシュホース法

アダムス・バッシュホース法のコード例(3段,3次精度)

微分方程式を範囲$[a, b]$でn分割して解く

```
double h = (b-a)/n; // 刻み幅
double x = a;  // xの初期値
double y = y0; // yの初期値
double fi, fi1 = 0, fi2 = 0;
for(int i = 0; i <= n-1; ++i){
    fi = func(x, y);
    if(i < 2){ // 最初の方はオイラー法を使う
        y = y+h*fi;
    }
    else{
        y = y+h*(5*fi2-16*fi1+23*fi)/12.0;
    }
    fi2 = fi1; fi1 = fi;
    x = x+h;    // xの更新
}
```

f_i, f_{i-1}, f_{i-2}を格納するための変数

$i < 2$の時はオイラー法で更新(アダムス・バッシュホース法で計算するために必要なデータがたまっていない)

アダムス・バッシュホース法によるyの更新

アダムス・バッシュホース法

アダムス・バッシュホース法の実行結果例

$$\frac{dy}{dx} = 2xy \qquad \text{解 } y = y_0 e^{x^2}$$

範囲$[0, 1]$, 初期値$y_0 = 1$, 刻み幅$h = 0.1$で$y = 1$まで計算

```
[Adams-Bashforth(4)]
y(1) = 2.630358987,  error = 0.08792284145

[Adams-Bashforth(3)]
y(1) = 2.643797513,  error = 0.07448431571

[RK3]
y(1) = 2.7183378,  error = 5.597151256e-05
[Heun method]
y(1) = 2.709057014,  error = 0.009224814449
[Eular method]
y(1) = 2.334633363,  error = 0.3836484654
```

3段(3次精度)の場合, 精度はオイラー法よりはよいが, RK3,ホイン法よりは精度が低い. **3次精度にしては誤差が大きい?**

アダムス･バッシュホース法

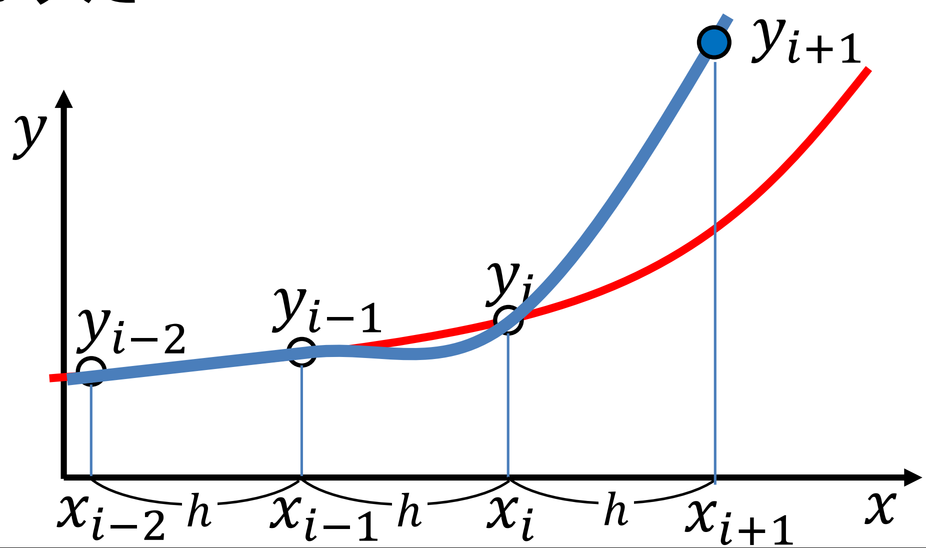
多数の点を使って補間しているのになぜ**精度が悪いのか?**

アダムス･バッシュホース法は**ラグランジュ補間多項式**に基づいている

⇒ 高次になると**ルンゲ現象**で端の方で**振動**する

$x_i, x_{i-1}, x_{i-2}, \ldots$での値を用いて$y(x_{i+1})$を計算している

⇒ **補外**を積み重ねることになるので特に**高次多項式になると誤差が大きい**

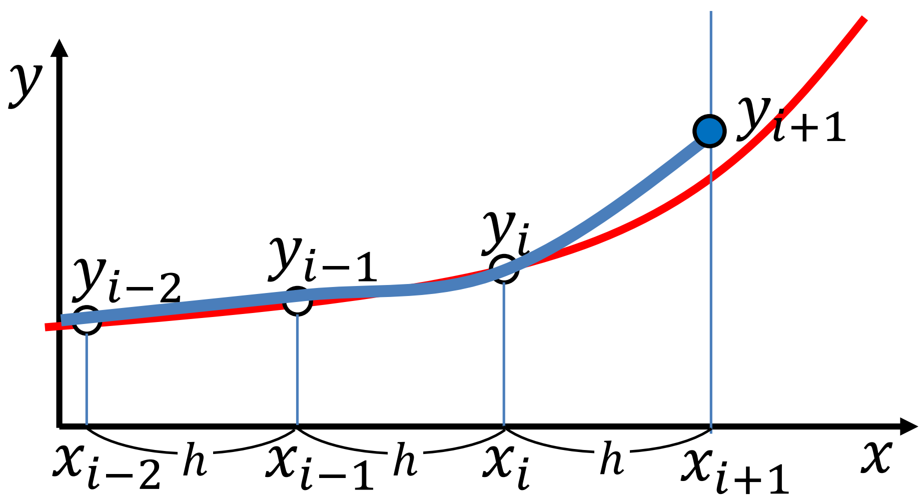


アダムス･ムルトン法

ホイン法の時と同様に**何らかの方法で**y_{i+1}**を求めて**, それを含めて$x_{i+1}, x_i, x_{i-1}, x_{i-2}, \dots$での値で$y_{i+1}$を**修正**する形にすれば**補外ではなく補間になる**

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

アダムス･ムルトン法



アダムス·ムルトン法

アダムス·バッシュホース法と同様に, 積分で表した式の$f(x,y)$をx_{i+1}, y_{i+1}を含めてラグランジュ補間多項式で近似

$$y_{i+1} = y_i + \int_{x_i}^{x_{i+1}} f(x,y)dx$$

$f(x,y)$を$(x_{i+1}, y_{i+1}), (x_i, y_i), (x_{i-1}, y_{i-1})$ の3点で近似してみる

$$f(x,y) \approx \sum_{k=i-1}^{i+1} \prod_{j=i-1\sim i+1, j\neq k} \frac{x - x_j}{x_k - x_j} f_k$$

ここで $f_i = f(x_i, y_i)$

$$= \frac{x - x_i}{\underbrace{x_{i-1} - x_i}_{-h}} \frac{x - x_{i+1}}{\underbrace{x_{i-1} - x_{i+1}}_{-2h}} f_{i-1} + \frac{x - x_{i-1}}{\underbrace{x_i - x_{i-1}}_{h}} \frac{x - x_{i+1}}{\underbrace{x_i - x_{i+1}}_{-h}} f_i + \frac{x - x_{i-1}}{\underbrace{x_{i+1} - x_{i-1}}_{2h}} \frac{x - x_i}{\underbrace{x_{i+1} - x_i}_{h}} f_{i+1}$$

アダムス·バッシュホース法と同様に分母をhで表して積分

アダムス・ムルトン法

積分の式に代入すると:

$$y_{i+1} = y_i + \frac{f_{i-1}}{2h^2}\int_{x_i}^{x_{i+1}}(x-x_i)(x-x_{i+1})dx - \frac{f_i}{h^2}\int_{x_i}^{x_{i+1}}(x-x_{i-1})(x-x_{i+1})dx$$

$$+\frac{f_{i+1}}{2h^2}\int_{x_i}^{x_{i+1}}(x-x_{i-1})(x-x_i)dx$$

⇩ それぞれ展開して積分を計算

$$y_{i+1} = y_i + h\left(-\frac{1}{12}f_{i-1} + \frac{8}{12}f_i + \frac{5}{12}f_{i+1}\right)$$

2次以外にも1次や3次の場合でも近似式ができる:

$m = 1$ (2点) : $y_{i+1} = y_i + h\left(\frac{1}{2}f_i + \frac{1}{2}f_{i+1}\right)$ **:ホイン法(改良オイラー法)**

$m = 3$ (4点) : $y_{i+1} = y_i + h\left(\frac{1}{24}f_{i-2} - \frac{5}{24}f_{i-1} + \frac{19}{24}f_i + \frac{9}{24}f_{i+1}\right)$

予測子修正子法

ホイン法の時と同様に**何らかの方法で**y_{i+1}**を求めて**, それを含めて$x_{i+1}, x_i, x_{i-1}, x_{i-2}, \ldots$での値で$y_{i+1}$を**修正**する

「何らかの方法」をアダムス・バッシュホース法とすると以下の手順で計算できる

1. アダムス・バッシュホース法でy_{i+1}の**予測値**Y_{i+1}を計算
2. $f_{i+1} = f(x_i + h, Y_{i+1})$を計算
3. アダムス・ムルトン法でy_{i+1}の値を**修正**

予測子修正子法

予測子修正子法

アダムス・バッシュホース法+アダムス・ムルトン法の組み合わせだけでなく**他の方法**でも予測子修正子法を用いることで**精度を向上させることができる**

例) オイラー法+改良オイラー法

$$Y_{i+1} = y_i + hf(x_i, y_i)$$:前進オイラー法*

$$y_{i+1} = y_i + \frac{h}{2}\left(f(x_i, y_i) + f(x_{i+1}, Y_{i+1})\right)$$:改良オイラー法

(ホイン法の手順そのもの. そのためホイン法は予測子修正子法といえる)

*この後出てくる後退オイラー法に対してオイラー法を前進オイラー法と呼ぶ

予測子修正子法

アダムス・バッシュホース/ムルトン法による予測子修正子法のコード例(3段,3次精度) 微分方程式を範囲$[a, b]$でn分割して解く

```
double h = (b-a)/n; // 刻み幅
double x = a;  // xの初期値
double y = y0; // yの初期値
double f[4] = { 0, 0, 0, 0 };
for(int i = 0; i <= n-1; ++i){
    f[2] = func(x, y);
    if(i < 2){ // 最初の方はオイラー法を使う
        // 前進オイラー法でy_(i+1)の予測値を計算
        f[3] = func(x+h, y+h*f[2]);
        // 改良オイラー法で解を修正
        y = y+h*(f[2]+f[3])/2.0;
    } else {
        // y_(i+1)の予測値を計算
        double y1 = y+h*(5*f[0]-16*f[1]+23*f[2])/12.0;
        f[3] = func(x+h, y1);
        // アダムス・ムルトン法で解を修正
        y = y+h*(-f[1]+8*f[2]+5*f[3])/12.0;
    }
    f[0] = f[1]; f[1] = f[2];
    x = x+h;    // xの更新
}
```

$f_{i-2}, f_{i-1}, f_i, f_{i+1}$を格納するための配列. 格納順はここに書いた順番通り(f[0]がf_{i-2},⋯,f[3]がf_{i+1})

$i < 2$の時はオイラー法+改良オイラー法で更新

(アダムス・バッシュホース法で計算するために必要なデータがたまっていない)

アダムス・バッシュホース法による予測値Y_{i+1}の計算

アダムス・ムルトン法による解の修正の計算

予測子修正子法

予測子修正子法の実行結果例 $\dfrac{dy}{dx} = 2xy$ 解 $y = y_0 e^{x^2}$

範囲$[0, 1]$, 初期値$y_0 = 1$, 刻み幅$h = 0.1$で$y = 1$まで計算

```
[Adams-Bashforth(4) + Adams-Moulton(4)]
y(1) = 2.718022765,  error = 0.000259063212
[Adams-Bashforth(3) + Adams-Moulton(3)]
y(1) = 2.719505483,  error = 0.001223654434
[Forward Eular + Modified Eular]
y(1) = 2.709057014,  error = 0.009224814449

[RK4] y(1) = 2.718270175,  error = 1.165307551e-05
[RK3] y(1) = 2.7183378,  error = 5.597151256e-05
[Heun method] y(1) = 2.709057014,  error = 0.009224814449
[Adams-Bashforth(3)] y(1) = 2.643797513,  error = 0.07448431571
[Adams-Bashforth(4)] y(1) = 2.630358987,  error = 0.08792284145
```

アダムス・バッシュホース法単体より**1〜2桁精度は上**. ただし, 3次+3次もしくは4次+4次精度でもホイン法以上RK3以下. その代わり1ステップでの$f(x, y)$の計算回数はRK3の3回から2回に減っている(**計算速度は速い**)

今回の講義内容

- 今日の問題
- オイラー法, ホイン法, ルンゲ・クッタ法
- アダムス・バッシュホース法, 予測子修正子法
- **後退オイラー法**
- 連立常微分方程式, 高階常微分方程式

陰解法

アダムス・ムルトン法, 改良オイラー法などyの更新式にx_{i+1}での値が含まれる時に, **予測値などを使わずに直接解く**方法を**陰的解法(implicit method)**という

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

後退差分法

$$y_{i+1} = y_i + h\big((1-\alpha)f(x_i, y_i) + \alpha f(x_{i+1}, y_{i+1})\big)$$

ただし, $0 \le \alpha \le 1$.

$\alpha = 0$で前進オイラー法, $\alpha = \frac{1}{2}$で改良オイラー法(偏微分方程式に適用する場合はクランク・ニコルソン法という)

x_iまでの値を使うこれまで説明した方法は**陽的解法(explicit method)**という

後退オイラー法

前ページの式で$\alpha = 1$の時, **後退オイラー法** と呼ぶ

$$y_{i+1} = y_i + hf(x_{i+1}, y_{i+1})$$

x_i段階での未知数y_{i+1}が左辺にあり, **そのままでは解けない.**
⇒ どうやってy_{i+1}を求めるのか？

[方法1:直接代数方程式を解く]

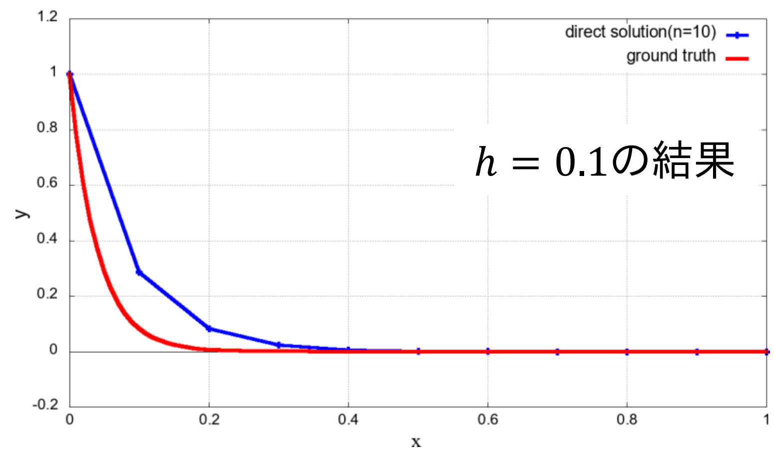
例) 実行結果例で使った硬い方程式: $\dfrac{dy}{dx} = -25y$

$$y_{i+1} = y_i - 25hy_{i+1}$$

$$\Downarrow$$

$$y_{i+1} = \frac{y_i}{1 + 25h}$$

(解析解は $y = y_0 e^{-25x}$)



後退オイラー法

[方法2:代数方程式をニュートン法で解く]

後退オイラー式を変形すると:

$$g(y_{i+1}) = y_{i+1} - hf(x_{i+1}, y_{i+1}) - y_i = 0$$

$g(y_{i+1})$の根を**ニュートン法**で求めることで近似解が得られる

$$y_{i+1}^{(k+1)} = y_{i+1}^{(k)} - \frac{y_{i+1}^{(k)} - hf\left(x_{i+1}, y_{i+1}^{(k)}\right) - y_i}{1 - h\dfrac{df}{dy_{i+1}}\left(x_{i+1}, y_{i+1}^{(k)}\right)} \qquad \text{ただし, } y_{i+1}^{(0)} = y_i$$

(ニュートン法の更新式$y^{(k)} = y^{(k)} - g/g'$に上の式を当てはめたもの)

[方法3:代数方程式を反復法で解く]

線形システムを解いたときのように反復的にy_{i+1}を更新して解く
(この方法は主に偏微分方程式の時に使われる)

$$y_{i+1}^{(k+1)} = y_i + hf\left(x_{i+1}, y_{i+1}^{(k)}\right)$$

後退オイラー法

後退オイラー法のコード例(方法2のニュートン法を用いたもの)

微分方程式を範囲$[a, b]$でn分割して解く

```
double h = (b-a)/n; // 刻み幅
double x = a;  // xの初期値
double y = y0; // yの初期値
for(int i = 0; i <= n-1; ++i){
    // ニュートン法でy_(i+1)を求める
    double yk = y; // y_(i+1)計算用の一時変数
    for(int k = 0; k < max_iter; ++k){
        double g = yk-h*func(x+h, yk)-y;
        double gf = 1-h*dfunc(x+h, yk);
        yk = yk-g/dg;
        if(fabs(g/dg) < eps || fabs(g) < eps) break;
    }
    y = yk; // yの更新
    x = x+h;    // xの更新
}
```

ニュートン法の更新式のために$g(y_{i+1})$と$g'(y_{i+1})$を計算

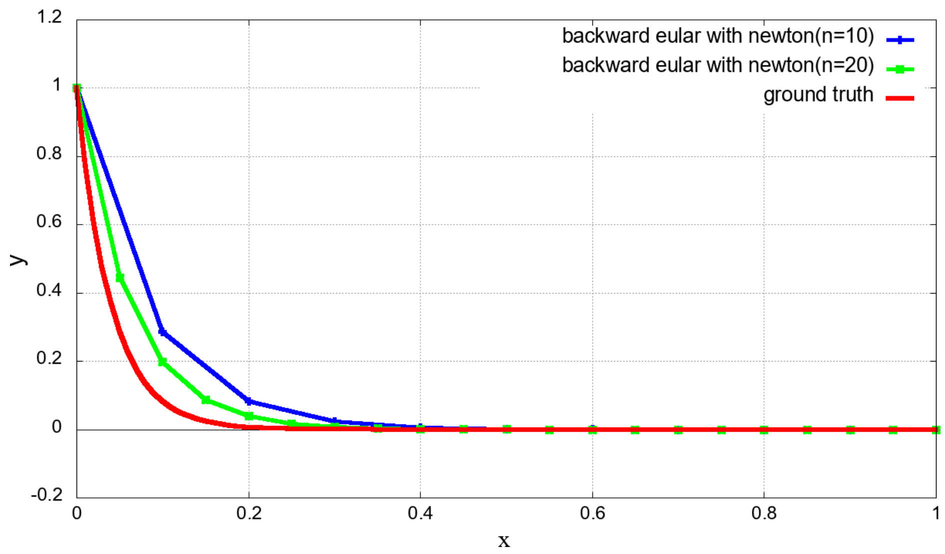
後退オイラー法

後退オイラー法の実行結果例(硬い方程式)

$$\frac{dy}{dx} = -25y \quad \text{解 } y = y_0 e^{-25x}$$

範囲$[0, 1]$, 初期値$y_0 = 1$, 刻み幅$h = 0.1$

```
y(0.1) = 0.2857142857
y(0.2) = 0.08163265306
y(0.3) = 0.02332361516
y(0.4) = 0.006663890046
y(0.5) = 0.001903968585
y(0.6) = 0.0005439910241
y(0.7) = 0.0001554260069
y(0.8) = 4.440743054e-05
y(0.9) = 1.26878373e-05
y(1) = 3.625096371e-06

error = 3.625082483e-06        → y(1)の誤差
ground truth = 1.3887943e-11   → y(1)の真値
```



前進オイラー法で発散していた$h = 0.1$ $(n = 10)$でも安定して解が求められている.
ただし, 精度は1次なので高精度な解がほしいならばhを小さくするか, 他の方法を用いた方が良い

後退オイラー法

後退オイラー法について

- **後退オイラー法の精度**は前進オイラー法と同じ**1次精度** $O(h)$ (後退オイラー法は1段の陰的ルンゲ･クッタ法に相当)
- 前進オイラー法より**計算安定性は非常に高い**. 大きな刻み幅hでも解が振動して発散しない. 陰的解法はほとんどの場合安定性が高いため, **数値計算が不安定になってしまったときの解決策**として覚えておくとよい
- 実際にはこれはあまり使われず, 偏微分方程式を解くときなどは**クランク･ニコルソン法の方がよく用いられる.**

今回の講義内容

- 今日の問題
- オイラー法, ホイン法, ルンゲ･クッタ法
- アダムス･バッシュホース法, 予測子修正子法
- 後退オイラー法
- **連立常微分方程式, 高階常微分方程式**

連立常微分方程式

モデル化したい現象が1つの変数(y)だけで表すことができるとは限らない

⇒ 多くの現象は**複数の変数**で表される

複数の変数を$y_1, y_2, \dots, y_n$とすると連立常微分方程式は:

$$\begin{gathered} y_1'(x) = f_1(x, y_1, y_2, \dots, y_n) \\ y_2'(x) = f_2(x, y_1, y_2, \dots, y_n) \\ \vdots \\ y_n'(x) = f_n(x, y_1, y_2, \dots, y_n) \end{gathered}$$

解を求める範囲を$[a, b]$として初期値$y_1(a), y_2(a), \dots, y_n(a)$は与えられている場合にどうやって連立微分方程式を解くか?

連立常微分方程式

$y_1, y_2, \dots, y_n$と$f_1, f_2, \dots, f_n$をベクトルで表記してみよう

$$\boldsymbol{y}(x) = \begin{pmatrix} y_1(x) \\ y_2(x) \\ \vdots \\ y_n(x) \end{pmatrix} \qquad \boldsymbol{f}(x, \boldsymbol{y}) = \begin{pmatrix} f_1(x, \boldsymbol{y}) \\ f_2(x, \boldsymbol{y}) \\ \vdots \\ f_n(x, \boldsymbol{y}) \end{pmatrix}$$

連立微分方程式： $\boldsymbol{y}'(x) = \boldsymbol{f}(x, \boldsymbol{y})$

ベクトルになっただけで式の形としては**これまでと同じ**なので, 例えば前進オイラー法なら：

$$\boldsymbol{y}_{i+1} = \boldsymbol{y}_i + h\boldsymbol{f}(x_i, \boldsymbol{y}_i)$$

として, 初期値$\boldsymbol{y}_0 = \left(y_1(a), y_2(a), \dots, y_n(a)\right)^T$から順番に計算していけば近似解が得られる(ホイン法やRKでも同じ).

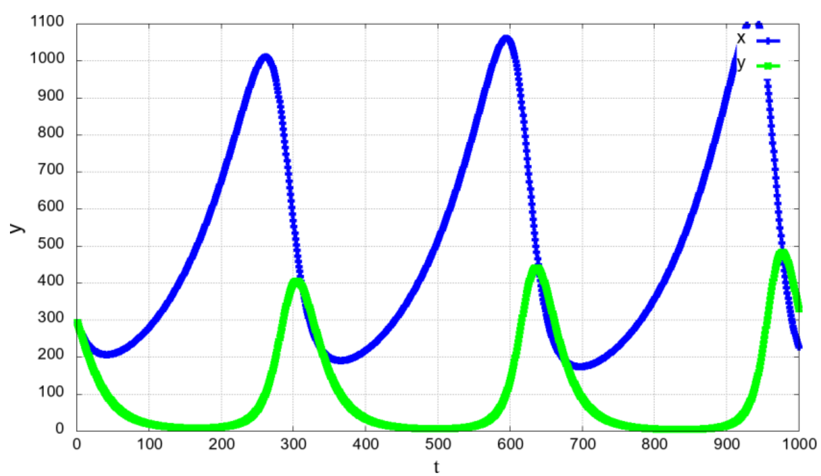
連立常微分方程式

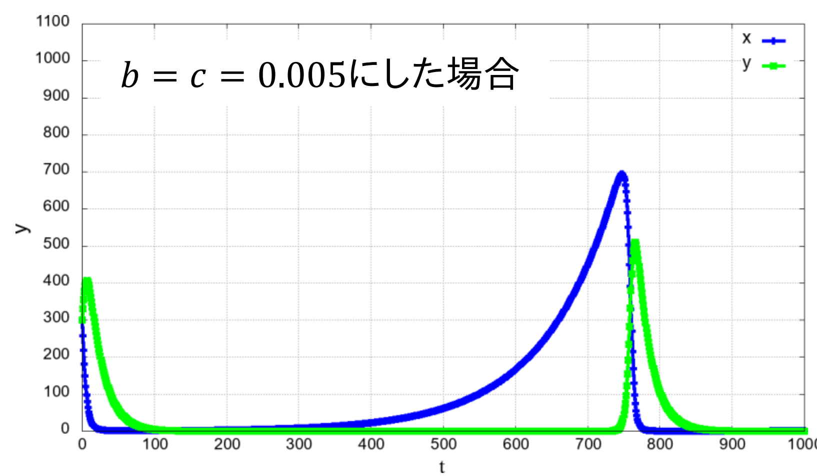
連立1階常微分方程式の例：**ロトカ・ヴォルテラの方程式**

$$\frac{dx}{dt} = \underbrace{ax}_{\text{被食者の自然増殖}} \underbrace{- bxy}_{\text{捕食されることによる減少}} \qquad \frac{dy}{dt} = \underbrace{cxy}_{\text{捕食による捕食者の増加}} \underbrace{- dy}_{\text{捕食者が増えたときの1匹当り食料減による減少}}$$

x：被食者の個体数
y：捕食者の個体数
t：時間

$x_0 = 300, y_0 = 300$として前進オイラー法でtを変化させたときのx, yの変化を計算
($h = 1$として$t = 0$日から1000日まで計算, $a = 0.01, d = 0.05, b = c = 0.0001$)





高階微分方程式

2次以上の微分が含まれる**高階微分方程式**の場合は？

$$\frac{d^n y}{dx^n} = f\left(x, y, \frac{dy}{dx}, \frac{d^2 y}{dx^2}, \dots, \frac{d^{n-1} y}{dx^{n-1}}\right)$$

$$\begin{cases} z_1(x) = y(x) \\ z_2(x) = \frac{dy}{dx}(x) \\ z_3(x) = \frac{d^2 y}{dx^2}(x) \\ \quad \vdots \\ z_n(x) = \frac{d^{n-1} y}{dx^{n-1}}(x) \end{cases} \text{と置くと} \begin{cases} \frac{dz_1}{dx}(x) = z_2(x) \\ \frac{dz_2}{dx}(x) = z_3(x) \\ \frac{dz_3}{dx}(x) = z_4(x) \\ \quad \vdots \\ \frac{dz_n}{dx}(x) = \frac{d^n y}{dx^n} = f(x, z_1, z_2, \dots, z_n) \end{cases}$$

このように書き換えれば結局**1階の連立微分方程式**として計算ができる ⇒ 先ほどと同じ方法で解ける

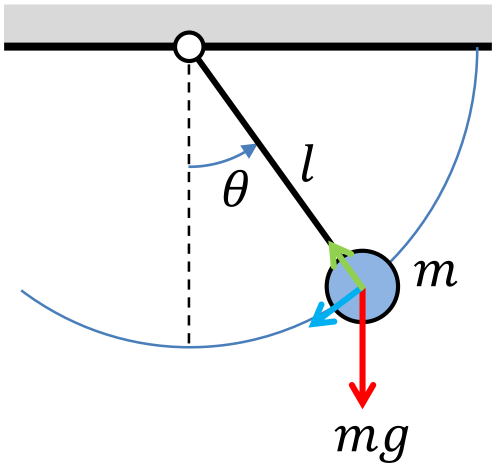
高階微分方程式

高階常微分方程式の例: **単振り子***

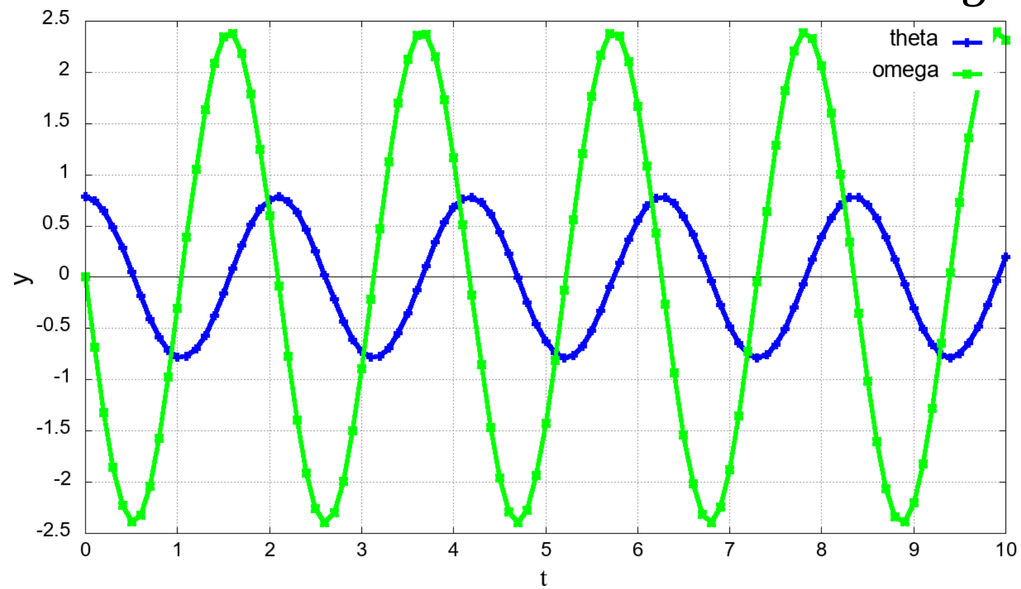
$$\frac{d^2\theta}{dt^2} = -gl\sin\theta \Rightarrow \begin{cases} \dfrac{d\theta}{dt} = \omega \\ \dfrac{d\omega}{dt} = -gl\sin\theta \end{cases}$$

$\dfrac{d\theta}{dt} = \omega$ (角速度)

とすると:



重力加速度$g = 9.8[m/s^2]$,
振り子の長さ$l = 1[m]$,
刻み幅$h = 0.1[s]$,
で$t = 0$から$10[s]$まで計算.
前進オイラーでは安定しなかったのでRK4を用いた



*単振り子を連ねた二重振り子というのもある

講義内容のまとめ

- 今日の問題：$\frac{dy}{dx} = f(x, y)$
- 1段階法(陽的解法)
  - オイラー法, ホイン法, ルンゲ・クッタ法
- 多段階法(陽的解法)
  - アダムス・バッシュホース法, 予測子修正子法
- 陰的解法
  - 後退オイラー法
- 連立常微分方程式, 高階常微分方程式

Appendix

(以降のページは補足資料です)

オイラー法の収束性

刻み幅hを0に近づけたときに誤差が0になれば，オイラー法の収束性が確かめられる: $\lim_{h\to 0} E_n = 0$

誤差は真値$y = g(x_i)$と近似値y_iの差なので

$g(x_{i+1}) = g(x_i + h)$のテイラー展開とオイラー法の適用($y_{i+1} = y_i + hf(x_i, y_i)$)

$$E_{i+1} = g(x_{i+1}) - y_{i+1}$$
$$= g(x_i) + hf\left(x_i, g(x_i)\right) + O(h^2) - y_i - hf(x_i, y_i)$$

$E_i = g(x_i) - y_i$および三角不等式$|x + y| \le |x| + |y|$より

$$|E_{i+1}| \le |E_n| + h\left|f\left(x_i, g(x_i)\right) - f(x_i, y_i)\right| + |O(h^2)| \quad \cdots ①$$

もし$f(x, y)$が**リプシッツ連続**ならば以下の**リプシッツ条件**を満たす$(0 \le L < 1)$

$$\left|f\left(x_i, g(x_i)\right) - f(x_i, y_i)\right| \le L|g(x_i) - y_i| = L|E_n|$$

これを①式に代入しても①式の不等号は変わらないので

$$|E_{i+1}| \le |E_n| + hL|E_n| + |O(h^2)|$$

$|O(h^2)|$を定数Cを用いてCh^2と書き換えると:

$$\boxed{|E_{i+1}| \le (1 + hL)|E_n| + Ch^2}$$

オイラー法の収束性

前ページ青枠の式で$E_0 = y(x_0) - y_0 = 0$であることから:

$$|E_1| \le Ch^2$$
$$|E_2| \le (1+hL)Ch^2 + Ch^2$$
$$|E_3| \le (1+hL)^2 Ch^2 + (1+hL)Ch^2 + Ch^2$$
$$\vdots$$
$$|E_n| \le Ch^2 \sum_{k=0}^{n-1} (1+hL)^k$$

等比数列の和は$(ar^n - a)/(r-1)$ (a:初項,r:公比)となるので

$$|E_n| \le \frac{Ch}{L}\left((1+hL)^n - 1\right)$$

不等式$1 + x \le e^x$および$h = (x_n - x_0)/n$より

$$\boxed{|E_n| \le \frac{Ch}{L}\left(e^{(x_n - x_0)L} - 1\right)}$$

計算範囲$[x_0, x_n]$が固定されているならば,
$h \to 0$で$|E_n| = 0$となることが分かる

⇒ オイラー法は$h \to 0$にすれば真値に収束することが証明できた

$y' = \lambda y$形の常微分方程式の例

$y' = \lambda y$の形の常微分方程式(解は$y = y_0 e^{\lambda x}$)は色々な現象をモデル化するのに使われている(ここで挙げたもの以外でも化学反応速度, コンデンサの電荷減衰, 振動減衰などいろいろある)

生物の個体数変化

食料量による生物の個体数変化

$$\frac{dy}{dt} = -\kappa y + \alpha$$

y : ある生物の個体数
κ :1個体の生存に必要な食料量
α :外部からの食糧供給量

光,X線などの関与媒質による吸収減衰

光やX線が雲や煙などの関与媒質内を通過するときの減衰

$$\frac{dI}{dx} = -\mu I$$

I : 光の強度
r : 減衰率

ヨウ素の核崩壊速度

放射性物質であるヨウ素が安定した核崩壊で安定したキセノンになる速度

$$\frac{dN}{dt} = -\lambda N$$

N : 要素の数
λ : 壊変定数(0.08664/日)

冷却速度

物体が周囲からの熱伝達で冷却されるときの温度変化(ニュートンの冷却の法則)

$$K\frac{dT}{dt} = -\sigma(T - T_{amb})$$

T : 温度, T_{amb}:周囲の温度(定数)
K, σ : 熱容量, 表面積/形状で決まる係数

2変数のテイラー展開

2変数関数$f(x, y)$がC^{n+1}連続であるならば, $f(x+h, y+k)$の(x, y)周りのテイラー展開は:

$$
\begin{aligned}
f(x+h, y+k) = f(x, y) &+ h\frac{\partial}{\partial x}f(x, y) + k\frac{\partial}{\partial y}f(x, y) \\
&+ \frac{h^2}{2!}\frac{\partial^2}{\partial x^2}f(x, y) + \frac{2hk}{2!}\frac{\partial^2}{\partial x \partial y}f(x, y) + \frac{k^2}{2!}\frac{\partial^2}{\partial y^2}f(x, y) + \cdots
\end{aligned}
$$

$$
\boxed{f(x+h, y+k) = \sum_{i=0}^{\infty}\frac{1}{i!}\left(h\frac{\partial}{\partial x} + k\frac{\partial}{\partial y}\right)^i f(x, y)}
$$

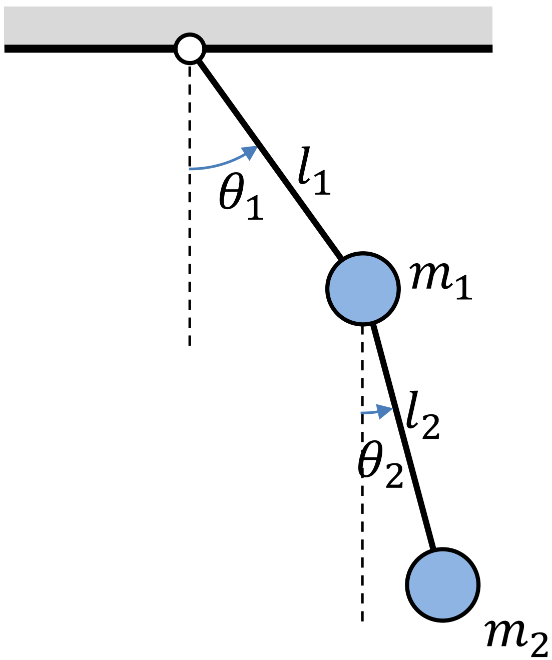
二重振り子

単振り子を2つ繋げた**二重振り子**は空気抵抗や摩擦を考えない場合でも解くことが難しい常微分方程式

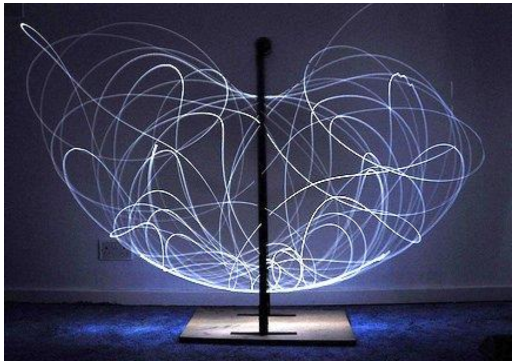
$$(m_1 + m_2) l_1 \ddot{\theta}_1 + m_2 l_2 \ddot{\theta}_2 \cos(\theta_1 - \theta_2) + m_2 l_2 \dot{\theta}_2^2 \sin(\theta_1 - \theta_2) + (m_1 + m_2) g \sin\theta_1 = 0$$

$$l_1 l_2 \ddot{\theta}_1 \cos(\theta_1 - \theta_2) + l_2^2 \ddot{\theta}_2 - l_1 l_2 \dot{\theta}_1^2 \sin(\theta_1 - \theta_2) + l_2 g \sin\theta_2 = 0$$

$\dot{\theta} = \frac{d\theta}{dt}$: 角速度, $\ddot{\theta} = \frac{d^2\theta}{dt^2}$: 角加速度



単純そうで実際にRK4とかで数値計算してみると動きの予測が付かない複雑な動きをするので**カオス現象**の例としてよく使われる



LEDと長時間露光で捉えた実際の二重振り子の動き
(Wikipedia:二重振り子より, CC BY 3.0)

自動微分について

コンピュータで微分を計算する方法として**自動微分(アルゴリズム的微分)**というものもある

大量の変数からなる複雑な数式を微分したい場合でも，**連鎖則**を使えば**単純な微分の組み合わせ**にできる

例) $z = x^2 y + \sqrt{x}$

$w_1 = x,\ w_2 = y,\ w_3 = w_1^2 w_2,\ w_4 = \sqrt{w_1}$とすると

zの式は $z = w_3 + w_4$ と表せる．

このときxもしくはyによる微分をまとめてz'と表すと

$z' = w_3' + w_4'$

$w_3' = 2w_1 w_2 \cdot w_1' + w_1^2 \cdot w_2'$

$w_4' = \frac{1}{2\sqrt{w_1}} \cdot {w_1}'$

xによる微分では $w_1' = \frac{\partial w_1}{\partial x} = 1, w_2' = \frac{\partial w_2}{\partial x} = 0$

yによる微分では $w_1' = \frac{\partial w_1}{\partial y} = 0, w_2' = \frac{\partial w_2}{\partial y} = 1$

$w_1' = 1, w_2' = 0$もしくは
$w_1' = 0, w_2' = 1$を
初期値として上方向に
計算を進めれば，
z'の式が求められる!

ボトムアップ型自動微分

自動微分について

逆に $z = w_3 + w_4$ なので,

$$\frac{\partial z}{\partial w_3} = 1, \frac{\partial z}{\partial w_4} = 1$$

$$\frac{\partial z}{\partial w_2} = \frac{\partial z}{\partial w_3}\frac{\partial w_3}{\partial w_2} + \frac{\partial z}{\partial w_4}\frac{\partial w_4}{\partial w_2} = w_1^2\frac{\partial z}{\partial w_3}$$

$$\frac{\partial z}{\partial w_1} = \frac{\partial z}{\partial w_3}\frac{\partial w_3}{\partial w_1} + \frac{\partial z}{\partial w_4}\frac{\partial w_4}{\partial w_1} = 2w_1 w_2\frac{\partial z}{\partial w_3} + \frac{1}{2\sqrt{w_1}}\frac{\partial z}{\partial w_4}$$

$\frac{\partial z}{\partial w_3} = 1, \frac{\partial z}{\partial w_4} = 1$から
スタートして下方向に
計算を進めれば,
z'が計算出来る!

トップダウン型自動微分

(ボトムアップ型/トップダウン型はそれぞれフォワードモード/リバースモードとも呼ばれる)

簡単な微分なら微分によってどういう数式になるのかが分かっているので,
それを**プログラミングすることは容易**

⇒ 実際には関数値を計算するプログラムコードから
微分を計算するコードへと自動変換して,
それを**導関数を用いた最適化**(最急降下法とか)などに用いる

あくまで**微分したい関数が分かっていることが前提**なので,
その関数そのものを求めたい微分方程式には使えない
(+離散データに基づく最適化の場合は数値微分がやはり必要)