電 301 数値解析
第 12 回
微分方程式の
数値解法 (2)

記号や用語および式の復習 (1)

- 解くべき微分方程式は $\dfrac{d\boldsymbol{x}}{dt} = \boldsymbol{f}(\boldsymbol{x}, t)$
- 取り扱うのは離散変数法 (時刻 $(t_0, t_1, \ldots, t_N)$ で微分方程式の数値解を計算)
- $h_k = t_{k+1} - t_k$ をステップ幅という．h_k が k によらず一定のときには h と書く．
- $\boldsymbol{x}(t_k)$ のことを $\boldsymbol{x}_k$ と書く．

記号や用語および式の復習 (2)

- 常微分方程式の数値解法:**1 段法** と **多段法**.
  段は英単語 **step** の訳.
- 1 段法の一般形:

  explicit(陽的) な公式 :

  $$\boldsymbol{\xi}_{k+1} = \boldsymbol{\xi}_k + h_k \boldsymbol{\Psi}(t_k, \boldsymbol{\xi}_k, h_k)$$

  implicit(陰的) な公式 :

  $$\boldsymbol{\xi}_{k+1} = \boldsymbol{\xi}_k + h_k \boldsymbol{\Psi}(t_k, t_{k+1}, \boldsymbol{\xi}_k, \boldsymbol{\xi}_{k+1}, h_k)$$

色々な常微分方程式の数値解法 (3)

- まず, Euler 法より精密な $\boldsymbol{\Psi}$ の構成法である Runge-Kutta 型公式について述べ (おもに explicit な公式), 続いて多段法について述べる.
- 参考文献は前回と同じなので再掲しない.
- 以下では, $\boldsymbol{x}(t)$ は $\dfrac{d\boldsymbol{x}}{dt} = \boldsymbol{f}(\boldsymbol{x}, t)$ の解とする.

Runge-Kutta 型公式 (1)

- Runge-Kutta 型公式は代表的な **1 段法**.
- 平均値の定理から, $\exists \boldsymbol{\Phi}\left(t_k, \boldsymbol{x}_k, h_k\right)$, $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + h_k \boldsymbol{\Phi}\left(t_k, \boldsymbol{x}_k, h_k\right)$. $\boldsymbol{\Phi}\left(t_k, \boldsymbol{x}_k, h_k\right)$ は未知なので, 既知の $\boldsymbol{\Psi}(t, \boldsymbol{x}(t), h_k)$ で近似する.
- $\boldsymbol{\Psi}(t, \boldsymbol{x}(t), h_k) - \boldsymbol{\Phi}(t, \boldsymbol{x}(t), h_k)$ を局所離散化誤差といい, $\boldsymbol{\tau}(t, r)$ と書く.

Runge-Kutta 型公式 (2)

- ある公式が $\exists C > 0,\ \exists \rho > 0,\ \forall r : 0 < r < \rho,$ $\forall t \in [a, b],\ \|\boldsymbol{\tau}(t, r)\|_\infty \leq C r^p$ を満たすとき, その公式を p 次精度の公式, あるいは p 次の公式という.

  区間の右端では, たとえば $\boldsymbol{x}(t+r) - \boldsymbol{x}(t)$ を $\boldsymbol{x}(t) - \boldsymbol{x}(t-r)$ で読み換えるなどといった, 若干の読み換えが必要.

Runge-Kutta 型公式 (3)

- ステップ幅が一定であるとき (h とする), p 次精度の公式に対し, ある $L > 0$ が存在して,

$$\forall k, \|\boldsymbol{x}_k - \boldsymbol{\xi}_k\| \leq \frac{e^{L(b-a)}}{L} C h^p$$

となることが示せる ([齊藤], pp. 225–226).

Runge-Kutta 型公式 (4)

- Runge-Kutta 型の公式は, いくつかの点における $\boldsymbol{f}(\boldsymbol{x}, t)$ の値を組み合わせて Ψ を構成することで p 次精度 $(p \geq 1)$ の公式を得る方法の総称.
- 色々なバリエーションがある.
- Heun 法は Runge-Kutta 法の一種.

Runge-Kutta 型公式 (5)

- Ψ を構成するために関数 $\boldsymbol{f}$ の値を s 個の点において評価するとき, これを s 段数 (s-stage) の Runge-Kutta 法と呼ぶ ([齊藤])
- s 段数の Runge-Kutta 法を構成する目的は, その公式が p 次精度となるようにすること.

Runge-Kutta 型公式 (6)

- $s \geq p$ である必要があることが示せるが ([齊藤]), $s = p$ とは限らない.
- 一般論の前に, 代表的な公式を紹介する (見方がわかればよく, 覚える必要はない).
- 差分方程式 $\boldsymbol{\xi}_{k+1} = \boldsymbol{\xi}_k + h_k \boldsymbol{\Psi}\left(t_k, \boldsymbol{\xi}_k, h_k\right)$ における関数 Ψ を p 次精度となるようにしたい.

Runge-Kutta 型公式 (7)

▷ 2 段数の Runge-Kutta 法

- 2 段数の Runge-Kutta 法は, $\boldsymbol{k}_1 = \boldsymbol{f}(\boldsymbol{\xi}_k, t_k)$, $\boldsymbol{k}_2 = \boldsymbol{f}(\boldsymbol{\xi}_k + h_k\beta\boldsymbol{k}_1, t_k + \alpha h_k)$ とし, $\Psi(t_k, \boldsymbol{\xi}_k) = c_1\boldsymbol{k}_1 + c_2\boldsymbol{k}_2$ とする公式である. α, β,c_1, c_2 をうまく選ぶと 2 次精度となる.

Runge-Kutta 型公式 (8)

- 2 段数の explicit な Runge-Kutta 法において, $\alpha > 0$ とし, $\beta = \alpha$, $c_1 = 1 - \frac{1}{2\alpha}$, $c_2 = \frac{1}{2\alpha}$ とすると, 2 次精度が達成できることが示せる. この意味で, 2 段数 2 次精度の explicit な Runge-Kutta 法には, 無限のバリエーションがある ([齊藤], pp. 227–229).
- よく知られた選び方を次ページに示す.

Runge-Kutta 型公式 (9)

2 段数 Runge-Kutta 法

	α	β	c_1	c_2
Heun 法	1	1	$\frac{1}{2}$	$\frac{1}{2}$
改良 Euler 法	$\frac{1}{2}$	$\frac{1}{2}$	0	1

Runge-Kutta 型公式 (10)

Runge-Kutta 法のパラメータは, 関数 f を形式的に Taylor 展開し, その低次の項が零になるように係数合わせをすることによって得られる. 計算は繁雑ではあるが, 機械的にできる. すべての Runge-Kutta 法を特徴付けることにはあまり意味がないので, 以下ではよく知られたパラメータの選び方のみをを紹介する. 以下で述べる, より高い段数の Runge-Kutta 法についても, 「なぜそんなパラメータを選ぶか」が一見して明らかでないと思われるが, それらは「辻褄が合うように係数合わせをして, うまくいったものの例」に過ぎない.

Runge-Kutta 型公式 (11)

▷ 4 段数の Runge-Kutta 法

- 4 段数の Runge-Kutta 法は,
  $\boldsymbol{k}_j = \boldsymbol{f}(\boldsymbol{\xi}_k + h_k \sum_{l<j} \beta_{jl} \boldsymbol{k}_l, t_k + \alpha_j h_k)$ (ただし $j = 1, \ldots, 4, \sum_{l<j} \beta_{jl} = \alpha_j$) とし, $\Psi(t_k, \boldsymbol{\xi}_k) = \sum_{j=1}^{4} c_j \boldsymbol{k}_j$(ただし $\sum_{j=1}^{4} c_j = 1$) とする公式である. ($j = 1$ のときは $\sum_{l<j} \cdots$ の項は $\cdots$ の部分にかかわらず零と定義する).

Runge-Kutta 型公式 (12)

Runge の 1/6 公式

α_1	α_2	α_3	α_4	c_1	c_2	c_3	c_4
0	$\frac{1}{2}$	$\frac{1}{2}$	1	$\frac{1}{6}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{6}$
$\beta_{21} = \frac{1}{2}, \beta_{32} = \frac{1}{2}, \beta_{43} = 1$							

記載のないパラメータは零

この公式は, RK4, RK41, 古典的 Runge-Kutta 法など, 色々の名前で呼ばれる.

Runge-Kutta 型公式 (13)

Kutta の 1/8 公式

α_1	α_2	α_3	α_4	c_1	c_2	c_3	c_4
0	$\frac{1}{3}$	$\frac{2}{3}$	1	$\frac{1}{8}$	$\frac{3}{8}$	$\frac{3}{8}$	$\frac{1}{8}$
$\beta_{21} = \frac{1}{3}, \beta_{31} = -\frac{1}{3}, \beta_{32} = 1$							
$\beta_{41} = 1, \beta_{42} = -1, \beta_{43} = 1$							

記載のないパラメータは零

Runge-Kutta 型公式 (14)

- 続いて, Runge-Kutta 法の一般形とその表記法について述べる.
- $\Psi(t_k, \boldsymbol{\xi}_k) = \sum_{j=1}^{s} c_j \boldsymbol{k}_j$, $\sum_{j=1}^{s} c_j = 1$ という構成を考える. $\boldsymbol{k}_j$ は後述. $j = 1, \ldots, s$ に注意.

Runge-Kutta 型公式 (15)

Explicit(陽的) : $\sum_{l<j}\beta_{jl} = \alpha_j$ としたとき,

$$\boldsymbol{k}_j = \boldsymbol{f}(\boldsymbol{\xi}_k + h_k \sum_{l<j} \beta_{jl}\boldsymbol{k}_l, t_k + \alpha_j h_k),$$

Implicit(陰的) : $\sum_{l=1}^{s}\beta_{jl} = \alpha_j$ としたとき,

$$\boldsymbol{k}_j = \boldsymbol{f}(\boldsymbol{\xi}_k + h_k \sum_{l=1}^{s} \beta_{jl}\boldsymbol{k}_l, t_k + \alpha_j h_k)$$

Runge-Kutta 型公式 (16)

- Euler 法は 1 段数の Runge-Kutta 法であると解釈できる.
- s 段数の Runge-Kutta 型公式は, s 個のパラメータ $\alpha_1, \ldots, \alpha_s$, s 個のパラメータ $c_1, \ldots, c_s$ および s^2 個のパラメータ (β_{il}) によって特徴付けられる. ただし, $\alpha_j = \sum_{l=1}^{s} \beta_{jl}$ である.

Runge-Kutta 型公式 (17)

- Runge-Kutta 型公式のパラメータを簡潔に表示するために, Butcher 配列という形式が用いられる. これは, 次のような表である.

$$
\begin{array}{c|cccc}
\alpha_1 & \beta_{11} & \beta_{12} & \cdots & \beta_{1s} \\
\alpha_2 & \beta_{21} & \beta_{22} & \cdots & \beta_{2s} \\
\vdots & \vdots & \vdots & & \vdots \\
\alpha_s & \beta_{s1} & \beta_{s2} & \cdots & \beta_{ss} \\
\hline
 & c_1 & c_2 & \cdots & c_s
\end{array}
$$

Runge-Kutta 型公式 (18)

- explicit な公式の Butcher 配列では, 右下がりの対角線とその上の要素は零である.
- $s = p = 2$, $s = p = 3$, $s = p = 4$ の場合は, パラメータを含んだ Runge-Kutta 型公式の一般形 (の Butcher 配列) を書き下すことができる ([Butcher]).

Runge-Kutta 型公式 (19)

- $s = p = 2$ とした場合の explicit な公式の一般形についてはすでに述べた.
- $s = p = 3$, $s = p = 4$ については, パラメータ値に関する場合分けが必要で繁雑なので, この講義では述べない ([Butcher] 参照).
- 以下, Butcher 配列の例を示す ([齊藤]).

1 段数 ($s = p = 1$) [齊藤]

前進 Euler 法

0	0
	1

後退 Euler 法

1	1
	1

2 段数 ($s = p = 2$) [齊藤]

Heun 法

0	0	0
1	1	0
	$\frac{1}{2}$	$\frac{1}{2}$

改良 Euler 法

0	0	0
$\frac{1}{2}$	$\frac{1}{2}$	0
	0	1

Crank–Nicolson 法

0	0	0
1	$\frac{1}{2}$	$\frac{1}{2}$
	$\frac{1}{2}$	$\frac{1}{2}$

3 段数 ($s = p = 3$) [齊藤]

Heun 法

0	0	0	0
$\frac{1}{4}$	$\frac{1}{4}$	0	0
$\frac{2}{3}$	$-\frac{2}{9}$	$\frac{8}{9}$	0
	$\frac{1}{4}$	0	$\frac{3}{4}$

Kutta 法

0	0	0	0
$\frac{1}{2}$	$\frac{1}{2}$	0	0
1	-1	2	0
	$\frac{1}{6}$	$\frac{2}{3}$	$\frac{1}{6}$

4 段数 ($s = p = 4$) [山本], [Hairer] Vol. 1

Runge の 1/6 公式

0	0	0	0	0
$\frac{1}{2}$	$\frac{1}{2}$	0	0	0
$\frac{1}{2}$	0	$\frac{1}{2}$	0	0
1	0	0	1	0
	$\frac{1}{6}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{6}$

Kutta の 1/8 公式

0	0	0	0	0
$\frac{1}{3}$	$\frac{1}{3}$	0	0	0
$\frac{2}{3}$	$-\frac{1}{3}$	1	0	0
1	1	-1	1	0
	$\frac{1}{8}$	$\frac{3}{8}$	$\frac{3}{8}$	$\frac{1}{8}$

Runge-Kutta 型公式 (23)

- $\boldsymbol{B} = (\beta_{jl})_{1\leq j\leq s, 1\leq l\leq s}$ を Butcher 配列の中央部にある行列, $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_s)^T$ を左側にあるベクトル, $\boldsymbol{c} = (c_1, \ldots, c_s)$ を下側にあるベクトルとし, 対応する Butcher 配列を $(\boldsymbol{B}, \boldsymbol{\alpha}, \boldsymbol{c})$ と表記する.

Runge-Kutta 型公式 (24)

- explicit な Runge-Kutta 法に関し, $p \geq 5$ なら p 段数では p 次精度は実現不可, $p \geq 7$ なら $p+1$ 段数では p 次精度は実現不可, $p \geq 8$ なら $p+2$ 段数では p 次精度は実現不可という事実が知られている ([Hairer et al], Vol. 1).
- 上記は指定された下限より大きい s に対して公式の存在を保証するわけではない.

Runge-Kutta 型公式 (25)

- 5 次精度については 6 段数, 6 次精度については 7 段数, 8 次精度については 11 段数, 10 次精度については 17 段数の公式が知られている ([Hairer et al], Vol. 1).
- 浮動小数点演算の誤差のため, 高次精度の公式がつねに実用的とは限らない.

ステップ幅と誤差の制御 (1)

- 今までの議論では, ステップ幅の決め方は議論されていなかった.

- 後に述べるように, Butcher 配列 $(\boldsymbol{B}, \boldsymbol{\alpha}, \boldsymbol{c})$ を持つ p+1 次精度の公式と, Butcher 配列 $(\boldsymbol{B}, \boldsymbol{\alpha}, \overline{\boldsymbol{c}})$ を持つ p 次精度の公式が存在すれば, これを使って誤差を見積ることができる.

ステップ幅と誤差の制御 (2)

- いくつかの $(p, p+1)$ の組み合わせについて, 上述のような Butcher 配列が存在することが知られている.

- このようになっているとき, s 段数 $p+1$ 次精度の公式に p 次精度の公式が埋め込まれていると表現する ([齊藤]).

ステップ幅と誤差の制御 (3)

- 上記のような Runge-Kutta 法の組み合わせを**埋め込み型 Runge-Kutta 法**と呼ぶ ([齊藤]). その用途は, 数値解に含まれる誤差の見積りである.
- $(\boldsymbol{B}, \boldsymbol{\alpha}, \boldsymbol{c})$ に対応する Runge-Kutta 法が生成する数値解を $(\boldsymbol{\xi}_k)$, $(\boldsymbol{B}, \boldsymbol{\alpha}, \overline{\boldsymbol{c}})$ に対応する Runge-Kutta 法が生成する数値解を $(\overline{\boldsymbol{\xi}}_k)$ とする.

ステップ幅と誤差の制御 (4)

- 詳細は省くが, $\|\boldsymbol{\xi}_k - \overline{\boldsymbol{\xi}}_k\|$ の値から, 数値解に含まれる誤差を見積ることができる (ただし, 近似を含むので, 精密な評価ではない).
- 誤差の見積りが一定以下になるまでステップ幅をどんどん小さくしてゆく手法をステップ幅の自動調整という.

ステップ幅と誤差の制御 (5)

- ステップ幅の自動調整により, 数値解の正しさを, ある程度保証することができる (近似が含まれるので厳密ではない). 詳細については [齊藤] を参照.
- Scilab に実装されている RKF45 は埋め込み型 Runge-Kutta 法で, 6 段数 5 次精度の公式に 4 次精度の公式が埋め込まれている.

implicit な Runge-Kutta 型公式

- この講義では今までほとんど述べなかったが, implicit な Runge-Kutta 法は硬い問題 (後述) や differential-algebraic equation を解くのに用いられる. ただし, 内部反復を必要とするため, 必ずしも使いやすいわけではない.
- implicit な Runge-Kutta 法に興味がある者は [Hairer et el.] や [Butcher] などを参照せよ.

硬い問題

- 線形微分方程式 $\frac{d}{dt}\boldsymbol{x} = \boldsymbol{A}\boldsymbol{x}$ において, 行列 $\boldsymbol{A}$ の固有値がすべて相異なり, それら実部がすべて負であるものとする.
- $\boldsymbol{A}$ の固有値のうち実部が最小のものを λ_m, 最大のものを λ_M とする.
- $|\mathrm{Re}\,\lambda_M|/|Re\lambda_m|$ を硬度比という.
- 硬度比が大きい問題を硬い問題という. 硬い問題は数値的に解きにくいことが知られている.

多段法 (1)

- 多段法は, $\boldsymbol{\xi}_{k+1}$ を構成するために, $\boldsymbol{\xi}_k$ だけでなく, 過去の数値解の系列 $(\boldsymbol{\xi}_{k-L}, \ldots, \boldsymbol{\xi}_k)$ に関連した情報を使う方法の総称 (復習).
- Runge–Kutta 法では, 複数の点での関数値の線形結合により高精度の公式を得ていたが, 多段法は, $(\boldsymbol{\xi}_{k-L}, \ldots, \boldsymbol{\xi}_k)$ を使うことで類似した効果を得ることを意図している.

多段法 (2)

- 記法の便宜上, $\boldsymbol{\xi}_{k+L}$ を $(\boldsymbol{\xi}_k, \ldots, \boldsymbol{\xi}_{k+L-1})$ に関連した情報を使って構成するという形で問題を考え直す.
- 簡単のために, この講義では, 多段法ではステップ幅は h に固定されているものとする (ステップ幅可変の場合については [Hairer et al.], Vol. 1 を参照).

多段法 (3)

- Runge–Kutta 型公式と比較すると, $\boldsymbol{f}(\boldsymbol{\xi}_k, t_k)$, $\ldots$, $\boldsymbol{f}(\boldsymbol{\xi}_{k+L}, t_{k+L})$ の線形結合で近似関数を構成するのが簡単:
  $\boldsymbol{\xi}_{k+L} = \boldsymbol{\xi}_k + h\left(\sum_{l=0}^{L} \beta_l^{(0)} \boldsymbol{f}(\boldsymbol{\xi}_{k+l}, t_{k+l})\right)$
- $\beta_l^{(0)}$ $(0 \leq l \leq L)$ はパラメータとして残す.
- $\boldsymbol{\xi}_k$ を左辺に移項して $\cdots$

多段法 (4)

- $\boldsymbol{\xi}_{k+L} - \boldsymbol{\xi}_k = h\left(\sum_{l=0}^{L} \beta_l^{(0)} \boldsymbol{f}(\boldsymbol{\xi}_{k+l}, t_{k+l})\right)$
- 左辺を $\{\boldsymbol{\xi}_{k+l}\}_{0\le l\le L}$ をすべて使う形に拡張すると, 線形多段法の一般形が得られる (次式).

$$\sum_{l=0}^{L} \alpha_l \boldsymbol{\xi}_{k+l} = h\left(\sum_{l=0}^{L} \beta_l \boldsymbol{f}(\boldsymbol{\xi}_{k+l}, t_{k+l})\right)$$

多段法 (5)

- 先の式の $\{\alpha_l\}_{0\leq l\leq L}$, $\{\beta_l\}_{0\leq l\leq L}$ はパラメータで, これらの取り方によって色々な公式が得られる.
- 上述一般形の中で, 等号の右辺に $\boldsymbol{\xi}_{k+L}$ が含まれるものを **implicit(陰的)** な公式, 右辺に $\boldsymbol{\xi}_{k+L}$ が含まれないものを **explicit(陽的)** な公式という. これらの特徴は 1 段法と同様.

多段法 (6)

- 多段法を非線形にまで (形式的に) 拡張することも可能, この形で一般形が述べられている本もあるので ([山本]), 念のため書いておく.
  $\sum_{l=0}^{L} \alpha_l \boldsymbol{\xi}_{k+l} = h\boldsymbol{\Psi}\left(t_k, \ldots, t_{k+L}, \boldsymbol{\xi}_k, \ldots, \boldsymbol{\xi}_{k+L}; h\right)$
- 以下で, 代表的な多段法をいくつか紹介するが, その前に注意点を述べる.

多段法 (7)

- 段数 L の多段法を使うときには, $\boldsymbol{\xi}_0, \ldots, \boldsymbol{\xi}_{L-1}$ を (1 段法等によって) 事前に計算する.
- (線形) 多段法のメリットは $\{\alpha_l\}_{0\le l\le L}$ や $\{\beta_l\}_{0\le l\le L}$ をうまく選ぶと高次精度の公式が得られること. 一方で, 安定性などに関して注意すべき点も多い.
- 多段法が安定で収束するための必要十分条件として, 以下のものが知られている ([森]): (1) 多項式 $\alpha_0 + \alpha_1 z + \cdots + \alpha_L z^L$ の根の絶対値がすべて 1 以下で, 絶対値 1 の根は単根, (2)$\sum_{l=0}^{L} \alpha_l = 0$, (3) $\sum_{l=0}^{l} l\alpha_l - \sum_{l=0}^{L} \beta_l = 0$.

予測子修正子法

- 多段法において, explicit な公式で仮の解を求め, それを implicit な公式に代入してから修正する方法を, 予測子修正子法という. また, この方法で使われる explicit な公式を予測子, implicit な公式を修正子という.
- explicit な解法と implicit な解法の組み合わせは色々.

Adams 型公式 (1)

- 微分方程式 $\frac{d\boldsymbol{x}}{dt} = \boldsymbol{f}(\boldsymbol{x}, t), \quad \boldsymbol{x}(t_{k+L-1}) = \boldsymbol{x}_{k+L-1}$ は次の積分方程式と等価.

$$\boldsymbol{x}_{k+L} - \boldsymbol{x}_{k+L-1} = \int_{t_{k+L-1}}^{t_{k+L}} \boldsymbol{f}(\boldsymbol{\varphi}(\tau, t_k, \boldsymbol{x}_k), \tau) d\tau$$

- しかし, この積分は $\boldsymbol{\varphi}(\tau, t_k, \boldsymbol{x}_k)$ が未知だから計算できない.

Adams 型公式 (2)

- そこで, $\boldsymbol{\varphi}(\tau, t_k, \boldsymbol{x}_k)$ を時間 τ に関する 1 変数関数と見做し, Lagrange 補間多項式で近似することを考える.
- 近似に $(t_k, \boldsymbol{x}_k), \ldots, (t_{k+L-1}, \boldsymbol{x}_{k+L-1})$ を使うと, $\boldsymbol{x}_{k+L} - \boldsymbol{x}_{k+L-1} \simeq \sum_{l=0}^{L-1} \beta_l \boldsymbol{f}(\boldsymbol{x}_{k+l}, t_l)$ という形に, $(t_k, \boldsymbol{x}_k), \ldots, (t_{k+L}, \boldsymbol{x}_{k+L})$ を使うと, $\boldsymbol{x}_{k+L} - \boldsymbol{x}_{k+L-1} \simeq \sum_{l=0}^{L} \beta_l \boldsymbol{f}(\boldsymbol{x}_{k+l}, t_l)$ という形になる.

Adams 型公式 (3)

- 前ページの $\boldsymbol{x}_j$ を $\boldsymbol{\xi}_j$ で置き換えると $(k \leq j \leq k+L)$, 多段法による常微分方程式の数値解法の公式が得られる. これを, **Adams 型公式**という.
- 右辺に $(\boldsymbol{\xi}_{k+L}, t_{k+L})$ を含まない公式を **Adams–Bashforth 公式**, これを含む公式を **Adams–Moulton 公式**という.

Adams 型公式 (4)

- Adams–Bashforth 公式 は explicit, Adams–Moulton 公式は implicit である.
- 右辺の係数 $\{\beta_l\}_{0\leq l\leq L}$ は, Lagrange 補間多項式を積分することで得られる. この係数は L の値だけで決まり, 問題によって変わるわけではないので, 事前に計算するか, 文献等の値をそのまま用いればよい.

Adams 型公式 (5)

例として, L=2 の場合の Adams–Bashforth 公式を計算してみる. t_k において $\boldsymbol{f}(\boldsymbol{\xi}_k, t_k)$, t_{k+1} において $\boldsymbol{f}(\boldsymbol{\xi}_{k+1}, t_k + 1)$ という値を取る Lagrange 補間多項式は, $\dfrac{t - t_{k+1}}{t_k - t_{k+1}} \boldsymbol{f}(\boldsymbol{\xi}_k, t_k) + \dfrac{t - t_k}{t_{k+1} - t_k} \boldsymbol{f}(\boldsymbol{\xi}_{k+1}, t_{k+1})$ であり, $t_{k+1} - t_k = h$ であることに注意すると, $\displaystyle\int_{t_{k+1}}^{t_{k+2}} \frac{t - t_{k+1}}{t_k - t_{k+1}} = -\frac{1}{2h} (t - t_{k+1})^2 \Big|_{t_{k+1}}^{t_{k+2}} = -\frac{h}{2}$, $\displaystyle\int_{t_{k+1}}^{t_{k+2}} \frac{t - t_k}{t_{k+1} - t_k} = \frac{1}{2h} (t - t_k)^2 \Big|_{t_{k+1}}^{t_{k+2}} = \frac{3h}{2}$ である. よって, L=2 の場合の Adams–Bashforth 公式は,

$$\boldsymbol{\xi}_{k+2} - \boldsymbol{\xi}_{k+1} = h \left(-\frac{1}{2} \boldsymbol{f}(\boldsymbol{\xi}_k, t_k) + \frac{3}{2} \boldsymbol{f}(\boldsymbol{\xi}_{k+1}, t_{k+1}) \right).$$

BDF 法 (1)

- BDF 法は Backward Difference Formula 法の略．この方法は implicit な解法の一種．
- BDF 法は硬い系向けの解法として広く使われている ([三井]).
- $\boldsymbol{\xi}_{k+L}$ を $(\boldsymbol{\xi}_k, \ldots, \boldsymbol{\xi}_{k+L-1})$ に関連した情報から構成するという多段法の問題設定に戻る．

BDF 法 (2)

- $\boldsymbol{\xi}_{k+L}$ を未定のままにしたまま, $(t_k, \boldsymbol{\xi}_k), (t_{k+1}, \boldsymbol{\xi}_{k+1}), \ldots, (t_{k+L}, \boldsymbol{\xi}_{k+L})$ を通る L 次の Lagrange 補間多項式を構成する. これを $\boldsymbol{p}(t)$ とする.
- ステップ幅が等間隔だったから, $t_{k+L-1} = t_{k+l} - h$, $t_{k+L-2} = t_{k+l} - 2h$, ... 等に注意する.

BDF 法 (3)

- $\boldsymbol{p}(t)$ が微分方程式の解の近似関数となるように $\boldsymbol{\xi}_{k+L}$ を決めたい.
- 決め方はひと通りというわけではないが, $\boldsymbol{p}(t)$ の導関数の $t = t_{k+L}$ における値が微分方程式 $\frac{d\boldsymbol{x}}{dt} = \boldsymbol{f}(\boldsymbol{x}, t)$ の右辺と一致するように決めるというのはひとつの考え方である.

BDF 法 (4)

- $\left.\dfrac{d}{dt}\boldsymbol{p}(t)\right|_{t=t_{k+L}} = \boldsymbol{f}(\boldsymbol{\xi}_{k+L}, t)$ となるように $\boldsymbol{\xi}_{k+L}$ を定める解法を **BDF 法**という.
- Lagrange 補間多項式を微分し, $t = t_{k+L}$ における値を評価することは, 代数的にできる (ステップ幅が等間隔であることに注意).

BDF 法 (5)

- いくつかの L に対する BDF 法は …

$$
\begin{aligned}
L=1: &\quad -\boldsymbol{\xi}_k + \boldsymbol{\xi}_{k+1} = h\boldsymbol{f}(\boldsymbol{\xi}_{k+1}, t_{k+1}) \\
L=2: &\quad \tfrac{1}{2}\boldsymbol{\xi}_k - 2\boldsymbol{\xi}_{k+1} + \tfrac{3}{2}\boldsymbol{\xi}_{k+2} = h\boldsymbol{f}(\boldsymbol{\xi}_{k+2}, t_{k+2}) \\
L=3: &\quad -\tfrac{1}{3}\boldsymbol{\xi}_k + \tfrac{3}{2}\boldsymbol{\xi}_{k+1} - 3\boldsymbol{\xi}_{k+2} + \tfrac{11}{6}\boldsymbol{\xi}_{k+3} = h\boldsymbol{f}(\boldsymbol{\xi}_{k+3}, t_{k+3})
\end{aligned}
$$

- BDF 法は $L > 6$ に対して不安定となることが知られている.

Scilab による実行例 (1)

微分方程式

$$\frac{d}{dt}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \quad \begin{pmatrix} x_1(0) \\ x_2(0) \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

を解く. 解は $x_1(t) = \cos t$, $x_2(t) = \sin t$. この微分方程式は硬い系ではないので, BDF 法などを使う利点はないが, すべての解法を試してみる.

```
//関数定義（全解法共通）
function dx=f(t,x)
  dx=[0 -1;1 0]*x;
endfunction

t0=0;
//時間の刻みは問題に合わせて要調整.
t=0:.1:100;
x0=[1;0];
```

ode のオプションで使う公式を指定できる.

```
y=ode(x0,t0,t,f); //デフォルト
y=ode("adams",x0,t0,t,f); //Adams 法
y=ode("stiff",x0,t0,t,f); //BDF 法
y=ode("rk",x0,t0,t,f);    //Runge-Kutta 法
y=ode("rkf",x0,t0,t,f);   //RKF45
```

デフォルト解法は Adams 型予測子・修正子法と BDF 法の自動切り換え

各解法の誤差 (数値解 $\boldsymbol{y}(t)$(2 行 1001 列の行列) と対応する真の解の差の Frobenius ノルム) を比較すると次の通り.

解法	誤差	計算時間
デフォルト	5.33×10^{-5}	0.031
Adams	5.78×10^{-5}	0.111
BDF 法	7.61×10^{-4}	0.062
Runge-Kutta 法	9.10×10^{-9}	0.811
RKF45	1.66×10^{-5}	0.062

Scilab による実行例 (5)

以下は文献 [三井] に掲載されている硬い系である.

$$\frac{d}{dt}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} -2 & 1 \\ 1998 & -1999 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} -\cos t \\ 1999\cos t - \sin t \end{pmatrix},$$

初期値を $x_1(0) = 1,\ x_2(0) = 2$ とする. 解は $x_1(t) = e^{-t}$, $x_2(t) = e^{-t} + \cos t$ である (真の解は Mathematica によって求めた).

```
//関数定義（全解法共通）
function dx=f(t,x)
  dx=[-2 1;1998 -1999]*x+..
    [-cos(t);1999*cos(t)-sin(t)];
endfunction
//注意: 記号.. は「次の行に続く」という意味
t0=0;
t=0:.5:100;
x0=[1;2];
```

公式を指定して解く方法は先ほどと同じ.

```
y=ode(x0,t0,t,f); //デフォルト
y=ode("adams",x0,t0,t,f); //Adams 法
y=ode("stiff",x0,t0,t,f); //BDF 法
y=ode("rk",x0,t0,t,f);    //Runge-Kutta 法
y=ode("rkf",x0,t0,t,f);   //RKF45
```

各解法の誤差 (数値解 $\boldsymbol{y}(t)$(2 行 1001 列の行列) と対応する真の解の差の Frobenius ノルム) を比較すると次の通り.

解法	誤差	計算時間
デフォルト	3.3×10^{-7}	0.031
Adams	2.1×10^{-6}	0.484
BDF 法	4.9×10^{-7}	0.109
Runge-Kutta 法	7.5×10^{-1}	29.547
RKF45	8.9×10^{-4}	2.028