電気 303 / 電情 303 数値解析 (11)

微分方程式の数値解法 (1)
常微分方程式とその分類・
離散変数法・オイラー法

今回および次回の参考文献

- 森正武, 数値解析, 第 2 版, 共立出版, 2002.
- 三井，常微分方程式の数値解法，岩波書店，2003.
- 山本, 数値解析入門 [増補版], サイエンス社, 2003.
- 笠原, 微分方程式の基礎, 朝倉書店, 1982.

- 齊藤宣一, 数値解析入門, 東京大学出版会, 2012.
- 伊理正夫, 藤野和建, 数値計算の常識, 共立出版, 1985.
- E. Hairer, S. P. Nørsett and G. Wanner, Solving Ordinary Differential Equations 1, 2/e, Springer, 1993.

- E. Hairer, S. P. Nørsett and G. Wanner, Solving Ordinary Differential Equations 2, 2/e, Springer, 1996.
- J. C. Butcher, Numerical Methods for Ordinary Differential Equations, Wiley, 2003.
- R. Riaza, DIfferential-Algebraic Systems, World Scientific, 2008.

微分方程式とは

- 未知関数とその (偏) 導関数のあいだに関係式が与えられたとき, これを**微分方程式**という (岩波数学入門辞典). 未知関数が 1 変数関数のとき**常微分方程式**, 多変数関数のとき**偏微分方程式**という (岩波数学入門辞典).
- 上記の関係式を満たす関数を求めることを, **微分方程式を解く**という (同上).

微分方程式の数値解法の必要性

- 物理現象は微分方程式で表現できる.
- たとえば人工衛星を静止軌道に打ち上げるなどといった応用問題を解くためには, ロケットの挙動の精密な数学モデル (微分方程式モデル) を作る必要がある.

- ロケットを静止軌道に打ち上げるためには推進装置や姿勢制御装置を適切に制御する必要がある．この制御の方法を試行錯誤によって決定しようとすると，失敗のたびにロケットが失われるから，危険でもあるし，費用がいくらあっても足りない．

- ロケットの数学モデルに基づいて数値シミュレー ションをおこない, うまくいきそうな制御方式を決定してから実機を用いた実験をおこなうことにすれば, 失敗の可能性は劇的に低下する.
- このようなことをおこなうためには, 微分方程式を解く必要がある.

- 見方を変えると, 物理現象は微分方程式をリアルタイムで解いているようなものであり, そのシミュレー ションは, 微分方程式をコンピュータで解くことで物理現象の再現を試みるもの, と考えることもできる.
- 微分方程式は解析的に解けないことが一般的なので, 数値的な解法が必要になる.

- 第 11 回と 12 回では, 常微分方程式の数値解法を取り扱う.
- 第 13 回と 14 回では, 偏微分方程式の数値解法を取り扱う.
- 今回の講義では, 常微分方程式 (とその一般化) に絞って議論を進める.

常微分方程式の分類

- 以下では, 未知関数を x とし独立変数 (実数) を t とした常微分方程式を考える.
- 独立変数の物理的意味は何でもよいが, 直感的には t が時間の場合が理解しやすいので, 以下では t を時間あるいは時刻と呼ぶ. とはいっても, 以下の議論は t の物理的意味には依存しない.

- 微分方程式にパラメータが含まれることもある.
- パラメータは時間に依存しない定数のこともあるが…
- 入力があるシステムの挙動が微分方程式で表現されているとき, 入力はこの微分方程式のパラメータとして取り扱われる. このような場合には, パラメータは時間の関数である.

- 以下しばらくパラメータを含まない微分方程式を考える．未知関数はスカラーでもベクトルでもよい．

- 独立変数を t とし, 未知関数を $x(t)$ あるいは $\boldsymbol{x}(t)$ とする (未知関数がベクトル値のとき太字にする).

- 独立変数を x とし, 未知関数を $y(x)$ としている教科書も多いので注意すること.

- 未知関数の 1 階までの微分を含む常微分方程式 (系) の一般形は次ページの通り．ただし，n を未知関数の次元，m を式の数とする．当面，$m = n$ であることは仮定しない．

n	m	微分方程式 (系)
1	1	$g\left(t, x, \frac{dx}{dt}\right) = 0$
≥ 2	1	$g\left(t, \boldsymbol{x}, \frac{d\boldsymbol{x}}{dt}\right) = 0$
1	≥ 2	$\boldsymbol{g}\left(t, x, \frac{dx}{dt}\right) = \mathbf{0}$
≥ 2	≥ 2	$\boldsymbol{g}\left(t, \boldsymbol{x}, \frac{d\boldsymbol{x}}{dt}\right) = \mathbf{0}$

x あるいは $\boldsymbol{x}$ は未知関数, $\mathbf{0}$ は零ベクトル

- 次に, 未知関数の n 階微分までを含む微分方程式を考える.
- 以下では, 未知関数の n 階微分までを含む微分方程式が, 適切に変数等を定義することにより, 未知関数の 1 階微分までを含む連立微分方程式へと書き直せることを見る.

- 未知関数の n 階微分までを含む微分方程式の一般形 (ただし未知関数がスカラーで式が 1 個) は, 次のようになる.

$$h\left(t, x, \frac{dx}{dt}, \frac{d^2x}{dt^2}, \ldots, \frac{d^nx}{dt^n}\right) = 0$$

- 上述の微分方程式を 1 階の (連立) 微分方程式に書き直すために, $(x_1, \ldots, x_n)$ を以下のように定義する.

$$
\begin{aligned}
x_1 &= x, \\
x_2 &= \frac{dx}{dt}, \\
&\cdots, \\
x_n &= \frac{d^{n-1}x}{dt^{n-1}}
\end{aligned}
$$

- すると…

$$
\begin{aligned}
&\frac{dx_1}{dt} = x_2 \\
&\cdots \\
&\frac{dx_{n-1}}{dt} = x_n \\
&h\left(t, x_1, \ldots, x_n, \frac{dx_n}{dt}\right) = 0
\end{aligned}
$$

- $\boldsymbol{x} = (x_1, \ldots, x_n)^T$ と定義し,

$$\boldsymbol{g}\left(t, \boldsymbol{x}, \frac{d\boldsymbol{x}}{dt}\right) = \begin{pmatrix} \dfrac{dx_1}{dt} - x_2 \\ \vdots \\ \dfrac{dx_{n-1}}{dt} - x_n \\ h(t, x_1, \ldots, x_n, \dfrac{dx_n}{dt}) \end{pmatrix}$$

とおくと…

- 先の微分方程式は

$$\boldsymbol{g}\left(t, \boldsymbol{x}, \frac{d\boldsymbol{x}}{dt}\right) = \boldsymbol{0}$$

となる．

- 以上のように, 未知関数の n 階までの微分を含む微分方程式は, 未知関数の 1 階までの微分を含む常微分方程式 (ただし変数はベクトルで式が複数) に書き直される.

- $\boldsymbol{x}$ がベクトルで, 関数が

$$h\left(t, \boldsymbol{x}, \frac{d\boldsymbol{x}}{dt}, \ldots, \frac{d^n\boldsymbol{x}}{dt^n}\right) = 0$$

となっている場合についても, $\boldsymbol{x}$ の各成分に着目すれば, 同じように処理できる.

- したがって, 常微分方程式を理論的に取り扱うときには, 1 階の微分方程式と高階の微分方程式を分けて取り扱う必要はなく,

$$\boldsymbol{g}(t, \boldsymbol{x}, \frac{d\boldsymbol{x}}{dt}) = \boldsymbol{0}$$

という形から出発すれば十分である.

- 以下しばらく, $\boldsymbol{x}$ がベクトル, $\boldsymbol{g}(\cdot)$ がベクトル値関数であることを前提とし, 微分方程式などを

$$\boldsymbol{g}\left(t, \boldsymbol{x}, \frac{d\boldsymbol{x}}{dt}\right) = \mathbf{0}$$

というふうに書く.

- $\boldsymbol{x}$ がスカラーである場合や式が 1 個の場合には, $\boldsymbol{x}$ を x で, $\boldsymbol{g}$ などを g などで読み換えて考えること.

- 微分方程式 $\boldsymbol{g}\left(t, \boldsymbol{x}, \dfrac{d\boldsymbol{x}}{dt}\right) = \mathbf{0}$ が $\dfrac{d\boldsymbol{x}}{dt}$ の項について解けるとき, すなわち, 適切な関数 $\boldsymbol{f}$ を定めることで,

$$\frac{d\boldsymbol{x}}{dt} = \boldsymbol{f}\left(t, \boldsymbol{x}\right)$$

と変形 できるとき, この 微分方程式は **explicit (陽的)** である という.

- explicit でない微分方程式のことを **implicit** (陰的) であるという.

- 微分方程式が (微分を含まない) 非線形方程式と連立されているとき, すなわち

$$\boldsymbol{g}\left(t, \boldsymbol{x}, \frac{d\boldsymbol{x}}{dt}\right) = \boldsymbol{0},$$

$$\boldsymbol{h}\left(t, \boldsymbol{x}\right) = \boldsymbol{0}$$

となっているとき, これを 微分代数方程式 (系) という.

- 微分代数方程式 (系) をディスクリプタシステムと呼ぶこともある.

- implicit な微分方程式は, 必ず explicit な形に変形できるとは限らない. そして, 部分的に explicit な形に書き直すと, 微分代数方程式があらわれることがある.

- 後述するように, explicit な常微分方程式では, (一定の条件のもとで) 解の存在と一意性が保証される.
- これに対し, 微分代数系では, 解が存在しないこともあり得る.

- 微分方程式には, たとえば

$$\frac{d\boldsymbol{x}}{dt} = \boldsymbol{f}\left(t, \boldsymbol{x}, \boldsymbol{\lambda}\right)$$

のように, パラメータ $\boldsymbol{\lambda}$ が含まれる場合もある.

- この講義では微分代数系は取り扱わない．それにもかかわらずこれらの名前を挙げたのは，Scilab における微分方程式関連の関数に，微分代数系用のものが含まれるからである．

- 以下の議論では, explicit な形に書き直せる常微分方程式のみを検討の対象とし, かつ 高階の微分方程式を explicit な 1 階の微分方程式に直す操作はすでに終わっているものする. パラメータはあってもなくてもよい.

- すなわち, 以下の議論では, 微分方程式

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

あるいは

$$\frac{d\boldsymbol{x}}{dt} = \boldsymbol{f}(\boldsymbol{x}, t, \boldsymbol{\lambda})$$

のみを検討の対象とする.

- $\boldsymbol{x}$ および $\boldsymbol{f}$ はスカラーであってもよいが, $\boldsymbol{x}$ と $\boldsymbol{f}$ の次元は一致していなければならない.

解の存在と一意性

- 微分方程式

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

  の解を区間 $t \in [a, b]$ で求める問題を考える.

- a を初期時刻, b を終了時刻という.

- 前ページの微分方程式の解であって, $t = a$ において値が $\boldsymbol{x}_a$ となるものが存在するとき, これを

$$\boldsymbol{\varphi}(t, a, \boldsymbol{x}_a)$$

と書く.

- $\boldsymbol{\varphi}(t, a, \boldsymbol{x}_a)$ の存在性については後述.
- $\boldsymbol{\varphi}(t, a, \boldsymbol{x}_a)$ が存在したとしても, ある t に対して $\boldsymbol{\varphi}(t, a, \boldsymbol{x}_a)$ が一意的に定まらない場合には, これは時間の関数とはいえない. $\boldsymbol{\varphi}(t, a, \boldsymbol{x}_a)$ が一意的に定まるか否かについても後述.
- 上記の $\boldsymbol{x}_a$ を微分方程式の初期値という.

- 初期値 $\boldsymbol{x}_a$ に対応する $\boldsymbol{\varphi}(t, a, \boldsymbol{x}_a)$ を求める問題を, 微分方程式の初期値問題という.
- 初期値 $\boldsymbol{x}_0$ の成分の一部と $t = b$ において解が取るべき値の成分の一部が指定されていて, $t = a$ と $t = b$ において該当する成分が指定された値を取る $\boldsymbol{\varphi}(t, a, \boldsymbol{x}_0)$ を求める問題を, 微分方程式の境界値問題という.

- これから述べるように, 一定の条件の下で初期値問題は解を持つが, 境界値問題の解は必ずしも存在するとは限らない.

- 境界値問題を考えるのは, いくつかの重要な応用問題(たとえば最適制御) において境界値問題があらわれるからであるが, この講義では常微分方程式の境界値問題は取り扱わない.

- 以下では, 常微分方程式の解の存在に関する重要な定理をいくつか証明抜きで紹介する.

- 微分方程式の初期値問題

$$\frac{d\boldsymbol{x}}{dt} = \boldsymbol{f}(\boldsymbol{x}, t), \quad \boldsymbol{x}(t_0) = \boldsymbol{x}_0 \qquad (\Diamond)$$

を考える. ただし, $\boldsymbol{x}(t) \in \mathbb{R}^n$ であり,

$$\boldsymbol{f} : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^n$$

である.

- 初期時刻 t_0 および初期値 $\boldsymbol{x}_0$ を固定する.
- G を, $(\boldsymbol{x}_0, t_0)$ を含む開集合とする.

(A1) $\boldsymbol{f}$ は, G において 連続であると 仮定する.

(A2) $\boldsymbol{f}$ は, G において, t に関して一様に, $\boldsymbol{x}$ について Lipschitz 連続, すなわち

$$\exists K > 0, \forall(\boldsymbol{x}_1, t), (\boldsymbol{x}_2, t) \in G,$$
$$\|\boldsymbol{f}(t, \boldsymbol{x}_2) - f(t, \boldsymbol{x}_1)\| \leq K\|\boldsymbol{x}_2 - \boldsymbol{x}_1\|$$

と仮定する.

- **定理** (A1) および (A2) が成り立っているとき, 微分方程式の初期値問題 ($\Diamond$) は $(\boldsymbol{x}_0, t_0)$ のある近傍で解を持つ. ■

- 上記の定理で述べられている「ある近傍」は,定理の証明の仮定で構築される．この講義では明示しない．以降の定理についても同様.

- **定理** 先の条件のもとで, 微分方程式の 初期値問題 $(\diamondsuit)$ の 解はこの近傍において 一意的に定まる. ■

- **定理** 先の条件のもとで, 微分方程式の 初期値問題 $(\diamondsuit)$ の 解は初期値に関して 連続である. ■

- 次に, パラメータを含む微分方程式の初期値問題を考える.

$$\frac{d\boldsymbol{x}}{dt} = \boldsymbol{f}(t, \boldsymbol{x}, \boldsymbol{\lambda}), \quad \boldsymbol{x}(t_0) = \boldsymbol{x}_0 \qquad (\clubsuit)$$

ただし, $\boldsymbol{x}(t) \in \mathbb{R}^n$, $\boldsymbol{\lambda} \in \mathbb{R}^p$ $(p \in \mathbb{N})$ であり,

$$\boldsymbol{f} : \mathbb{R}^n \times \mathbb{R} \times \mathbb{R}^p \to \mathbb{R}^n$$

である.

- 初期時刻 t_0, 初期値 $\boldsymbol{x}_0$ およびパラメータ $\boldsymbol{\lambda}$ を固定する.
- G' を, $(\boldsymbol{x}_0, t_0, \boldsymbol{\lambda})$ を含む開集合とする.

(B1) $\boldsymbol{f}$ は, G' において 連続であると 仮定する.

(B2) $\boldsymbol{f}$ は, G' において, t および $\boldsymbol{\lambda}$ に関して一様に, $\boldsymbol{x}$ について Lipschitz 連続, すなわち

$$\exists K > 0, \forall (\boldsymbol{x}_1, t, \boldsymbol{\lambda}), (\boldsymbol{x}_2, t, \boldsymbol{\lambda}) \in G,$$

$$\|\boldsymbol{f}(t, \boldsymbol{x}_2, \boldsymbol{\lambda}) - f(t, \boldsymbol{x}_1, \boldsymbol{\lambda})\| \leq K \|\boldsymbol{x}_2 - \boldsymbol{x}_1\|$$

と仮定する.

- **定理**　(B1) および (B2) が成り立っているとき, 微分方程式の初期値問題 (♣) は $(\boldsymbol{x}_0, t_0, \boldsymbol{\lambda})$ のある近傍で解を持ち, その解は $\boldsymbol{\lambda}$ に関して連続である. ■

- 更に条件を追加すれば, 微分方程式の初期値問題 ($\clubsuit$) の解は $\boldsymbol{\lambda}$ について微分可能となる.

(C1) $\boldsymbol{f}$ は, G' において $\boldsymbol{x}$ および $\boldsymbol{\lambda}$ に関して 1 階連続微分可能であると仮定する.

(C2) $\dfrac{\partial \boldsymbol{f}}{\partial \boldsymbol{\lambda}}$ は G' において連続であると仮定する.

- **定理** (B1), (B2), (C1) および (C2) が成り立っているとき, 微分方程式の初期値問題 ($\clubsuit$) の解は $\boldsymbol{\lambda}$ に関して 1 階連続微分可能である. ■

- 先に述べたように, 高階の微分方程式は1階の(連立)微分方程式(系)に書き直せるので, 高階の微分方程式について改めて議論する必要はない.

- この講義では, 今後は, 解の存在と一意性が保証された微分方程式のみを検討の対象とする.

離散変数法

- この講義では, 未知関数 $\boldsymbol{x}(t) = \boldsymbol{\varphi}(t, t_0, \boldsymbol{x}_0)$ がベクトル値関数の場合をおもに取り扱う.

- 常微分方程式を数値解法で解くときには, $[a, b]$ を適当な分点 $a = t_0 < t_1 < \cdots < t_N = b$ で分割し, 各 t_k において未知関数の値 $\boldsymbol{x}(t_k)$ を求めるという方法が取られることが多い．この方法を離散変数法という (岩波数学辞典).

- $h_k = t_{k+1} - t_k$ と定義し, これをステップ幅あるいは刻み幅呼ぶ([山本]). $t_k + h_k = t_{k+1}$ となることに注意. ステップ幅が k によらず一定のときには, これを h と書く.

- 以下では, $\boldsymbol{x}_k = \boldsymbol{x}(t_k)$ と書く $(0 \leq k \leq N)$.
- 各 k における $\boldsymbol{x}_k$ の近似を $\boldsymbol{\xi}_k$ とする.

- 離散変数法は, 未知の (ベクトルの) 列

$$(\boldsymbol{x}_k)_{k=1,2,\ldots,N}$$

を近似する別の列

$$(\boldsymbol{\xi}_k)_{k=1,2,\ldots,N}$$

を構成する方法の総称である.

- 近似列 $(\boldsymbol{\xi}_k)_{k=1,2,\ldots,N}$ は, 多くの場合, 差分方程式を解くことによって構成される.

- $\boldsymbol{x}_k$, $\boldsymbol{\varphi}$, $\boldsymbol{f}$ の第 j 成分を $x_{k,j}$, φ_j, f_j と書く.

- 中間値の定理より, ある $c_j \in [t_k, t_{k+1}]$ に対し,

$$
\begin{aligned}
x_{k+1,j} - x_{k,j} &= \left.\frac{d}{dt}\varphi\right|_{t=c} h_k \\
&= f_j(\boldsymbol{\varphi}(c_j, t_k, \boldsymbol{x}_k), c_j) h_j
\end{aligned}
$$

である. c_j の値は $\boldsymbol{x}_k$, $\boldsymbol{x}_{k+1}$, t_k, t_{k+1} で決まるが, j が変われば値も変わることに注意する.

- 形式的に

$$\phi_j(t_k, t_{k+1}, \boldsymbol{x}_k, \boldsymbol{x}_{k+1}; h_k) := f_j(\boldsymbol{\varphi}(c_j, t_k, \boldsymbol{x}_k), c_j)$$

と書き (ただし c が未知なので $\phi_j(\cdot)$ も未知), ϕ_j を成分として持つベクトル値関数を $\boldsymbol{\Phi}$ とすると, 次式が成り立つ.

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + h_k \boldsymbol{\Phi}(t_k, t_{k+1}, \boldsymbol{x}_k, \boldsymbol{x}_{k+1}; h_k).$$

- $\boldsymbol{\Phi}$ は未知であることを再度注意する.
- $\boldsymbol{\Phi}$ を近似する関数を $\boldsymbol{\Psi}$ とすると,

$$\boldsymbol{x}_{k+1} \simeq \boldsymbol{x}_k + h_k \boldsymbol{\Psi}(t_k, t_{k+1}, \boldsymbol{x}_k, \boldsymbol{x}_{k+1}; h_k)$$

となる.

- 上記の $\boldsymbol{x}_{k+1}$, $\boldsymbol{x}_k$ を $\boldsymbol{\xi}_{k+1}$, $\boldsymbol{\xi}_k$ で置き換え, $\simeq$ を等号に変えたものが, 1 段法と呼ばれる公式の一般形 (次式) である.

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

- 前回と同様に, 公式という言葉は計算の方法あるいは計算の手順という意味で用いられている. この用語は数値解析の分野では標準的であるが, 数学における公式 (厳密に正しい式) とは意味が異なる. 数値解析における公式とは, その手順にしたがえばとりあえず計算だけはものであって, 計算結果が (どの程度) 正しいかについては, 別途検討が必要なのである.

- 先に述べた公式は両辺に $\boldsymbol{\xi}_{k+1}$ を含んでいる. このように, 両辺に同じ変数を含んだ公式を, **implicit(陰的)** な公式という.
- この公式で数値解を構成しようとすると, 各ステップで非線形方程式を数値的に解く必要が生じることがある. 非線形方程式を数値的に解くには一般には反復が必要となり, この部分を内部反復という.

- implicit な公式を使った方がうまく解ける問題が存在するため(**硬い (stiff)** という ([Hairer et al.])), この公式には存在意義があるが(硬い系の定義は文献によって異なる), 内部反復が必要であるため, 必ずしも使いやすいとはいえない.

- 近似関数 $\boldsymbol{\Psi}$ を構成する際に変数 t_{k+1} と $\boldsymbol{\xi}_{k+1}$ を使うのを止めれば, 内部反復の必要はなくなる. このようにした公式を **explicit(陽的)** な公式という. 一般形を次に示す.

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

- 近似関数 $\boldsymbol{\Psi}$ の構成法は様々であるが, 今回の講義では単純な形のもののみを紹介し, より精密な方法については次回に述べる.
- $\boldsymbol{\xi}_{k+1}$ 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 $(\boldsymbol{\xi}_{k-L}, \ldots, \boldsymbol{\xi}_k)$ および対応する関数値 $\left(\boldsymbol{f}(\boldsymbol{\xi}_{k-L}, t_{k-L}), \ldots, \boldsymbol{f}(\boldsymbol{\xi}_k, t_k)\right)$ を使う方法もある. このような方法を多段法という.

- 離散変数法の具体的な公式や多段法についてはは次回の講義にまわすこととし, 今回の講義では, 離散変数法の中でももっとも初等的である Euler 法についてのみ述べる.

Euler 法

- 常微分方程式の数値解法のうち, もっとも初等的なものが Euler 法である.

- **(前進)Euler 法**は, 微分方程式

$$\frac{d\boldsymbol{x}}{dt} = \boldsymbol{f}(\boldsymbol{x}, t) \qquad (\heartsuit)$$

の解を, 差分方程式

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

を解くことで構成する方法である. これは 1 段法で, explicit である.

- 上述の公式は, 微分

$$\frac{d\boldsymbol{x}}{dt}$$

を前進差分

$$\frac{\boldsymbol{\xi}_{k+1} - \boldsymbol{\xi}_k}{h}$$

で近似することで得られている．また, 先の中間値の定理を使った議論で $c = t_k$ として近似関数を構成した場合に相当する.

- h_k をどんどん小さくすると, Euler 法が生成する近似解は微分方程式 ($\heartsuit$) の解に収束することが証明できる ([Hairer et al.]).

- 1 段法には k が大きくなるにしたがって, 数値的な近似解と微分方程式の真の解のあいだの誤差が増大するという特徴があるが, Euler 法ではこの問題が顕著である. この意味で, 高精度の近似解が必要な場合には, Euler 法は必ずしも適しているとはいえない. 一方で, Euler 法には公式が簡単であるというメリットもある.

- Euler 法を改良したものに，**Heun 法**と呼ばれる方法がある．これは，

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

$$\boldsymbol{\xi}_{k+1} = \boldsymbol{\xi}_k + \frac{h}{2}\Big(\boldsymbol{f}(\boldsymbol{z}_{k+1}, t_{k+1}) + \boldsymbol{f}(\boldsymbol{\xi}_k, t_k)\Big)$$

とする方法である (文献によって呼び方が変わることがある).

- **後退 Euler 法**は, 微分方程式 (♡) の解を, 差分方程式

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

により構成する方法である．implicit であること以外は前進 Euler 法と同じである．中間値の定理を使った議論で $c = t_{k+1}$ とした場合に相当する．

Scilab の常微分方程式関連関数一覧

`ode` ：1 階常微分方程式の初期値問題を解く.

`bvode` ：常微分方程式の境界値問題を解く.

`dae`, `daskr`, `dasrt`: 微分代数系を解く.

`odedc` ：ハイブリッドシステムを解く.

- この講義では `ode` 以外は取り扱わない.
- `ode` では様々な解法を指定することができ, その一部は次回の講義で述べるが, 今回の講義では Euler 法のみを利用する.

Scilab における Euler 法

- Scilab の関数 `ode` はオプションに Euler 法を含んでいない.
- Scilab の関数 `ode` において, オプションに含まれていない公式を利用するときは, `ode` の第一引数を `"discrete"` とし, 公式を自分で書き下す. この方法は, 1 段法であれば何でも実行できる方法である.

- 次回の講義で述べるが, 関数 `ode` で オプションに含まれる公式を利用する場合の使い方は今回の講義で述べる使い方とは異なるので, 注意せよ.

- 例として, 以下の線形微分方程式を解く問題を考える.

$$\frac{d\boldsymbol{x}}{dt} = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \boldsymbol{x}, \quad \boldsymbol{x}(0) = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \qquad (\sharp)$$

- 微分方程式 ($\sharp$) は解析的に解け, その解 (厳密解) は

$$x_1(t) = \cos t$$
$$x_2(t) = \sin t$$

である.

- 厳密解と 前進 Euler 法, Heun 法, 後退 Euler 法による近似解 (ステップ幅 h) を比較する.

- 行列 $\boldsymbol{A}$ を以下のように定義する.

$$\boldsymbol{A} = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$$

また, $\boldsymbol{I}$ を単位行列とする.

- この例に 前進 Euler 法, Heun 法, 後退 Euler 法の公式を適用すると, 以下の差分方程式が得られる.

  ▷ **前進 Euler 法**:

$$\boldsymbol{\xi}_{k+1} = (\boldsymbol{I} + h\boldsymbol{A})\boldsymbol{\xi}_k$$

▷ **Heun 法:**

$$\boldsymbol{z}_{k+1} = (\boldsymbol{I} + h\boldsymbol{A})\boldsymbol{\xi}_k,$$

$$\boldsymbol{\xi}_{k+1} = \boldsymbol{\xi}_k + \frac{h}{2}\Big(\boldsymbol{A}(\boldsymbol{\xi}_k + \boldsymbol{z}_{k+1})\Big)$$

▷ **後退 Euler 法:**

$$\boldsymbol{\xi}_{k+1} = \frac{1}{1+h^2}\left(\boldsymbol{I} + h\boldsymbol{A}\right)\boldsymbol{\xi}_k$$

- 次ページ以降に, $h = 0.1$, $t \in [0, 20]$ とした前進 Euler 法, 後退 Euler 法, Heun 法のプログラムを示す.

```
//前進 Euler 法
h=0.1;
deff('y=f(t,x)','y=[1 -h;h 1]*x');
x0=[1;0];
tLen=200;
x=ode("discrete",x0,1,1:tLen,f);
```

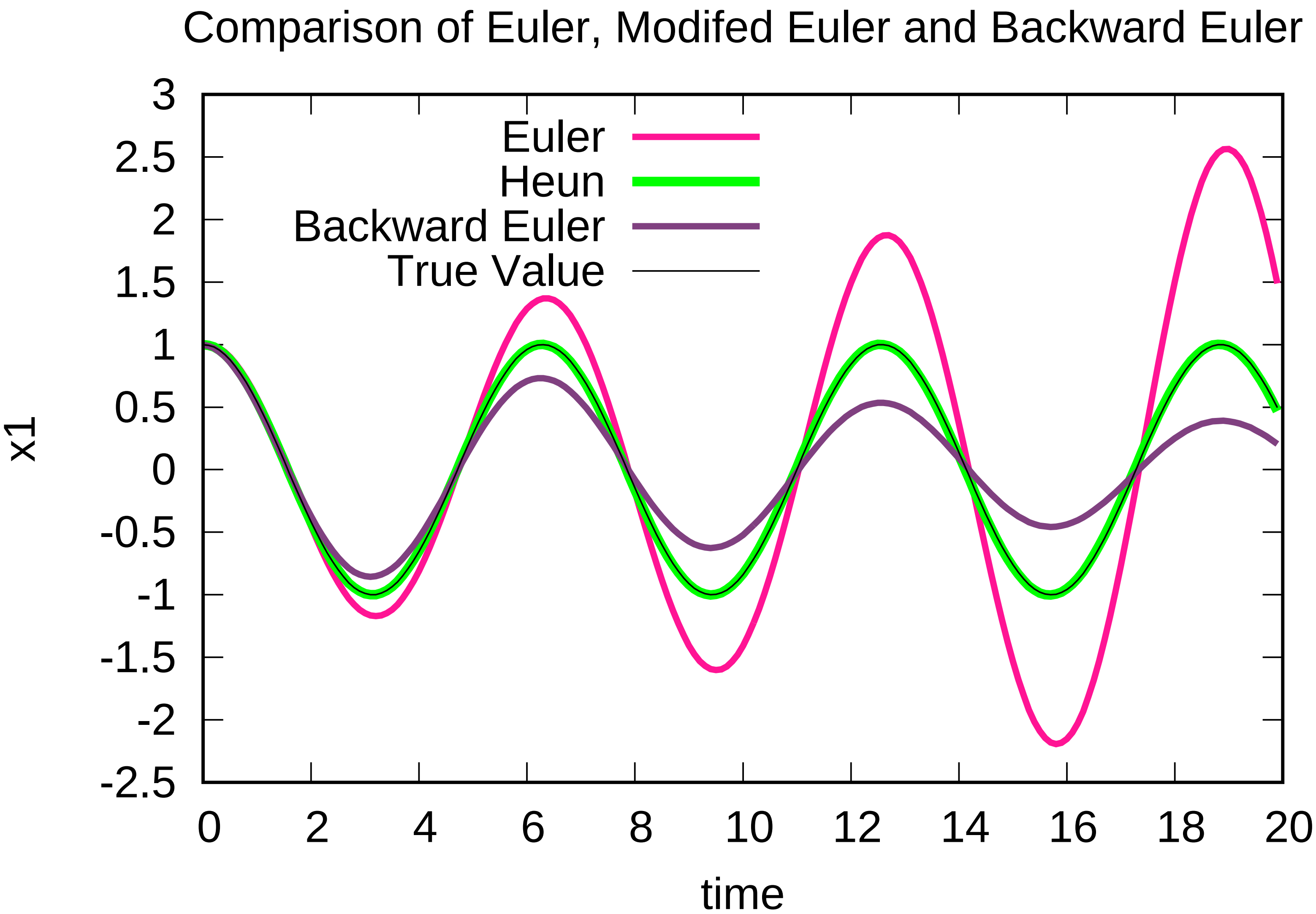
```
//後退 Euler 法
h=0.1;
deff('y=f(t,x)','y=([1 -h;h 1]*x)/(1+h^2)');
x0=[1;0];
tLen=200;
x=ode("discrete",x0,1,1:tLen,f);
```

```
//Heun 法
h=0.1;
deff('y=g(x)','y=[0 -1;1 0]*x');
function y=f(t,x)
w=g(x);
z=x+h*w;
y=x+h*(w+g(z))/2;
endfunction

x0=[1;0];
tLen=200;
x=ode("discrete",x0,1,1:tLen,f);
```

- 上記の違いは `y=f(t,x)` の内容で，ここで各公式に対応する関数が定義されている．

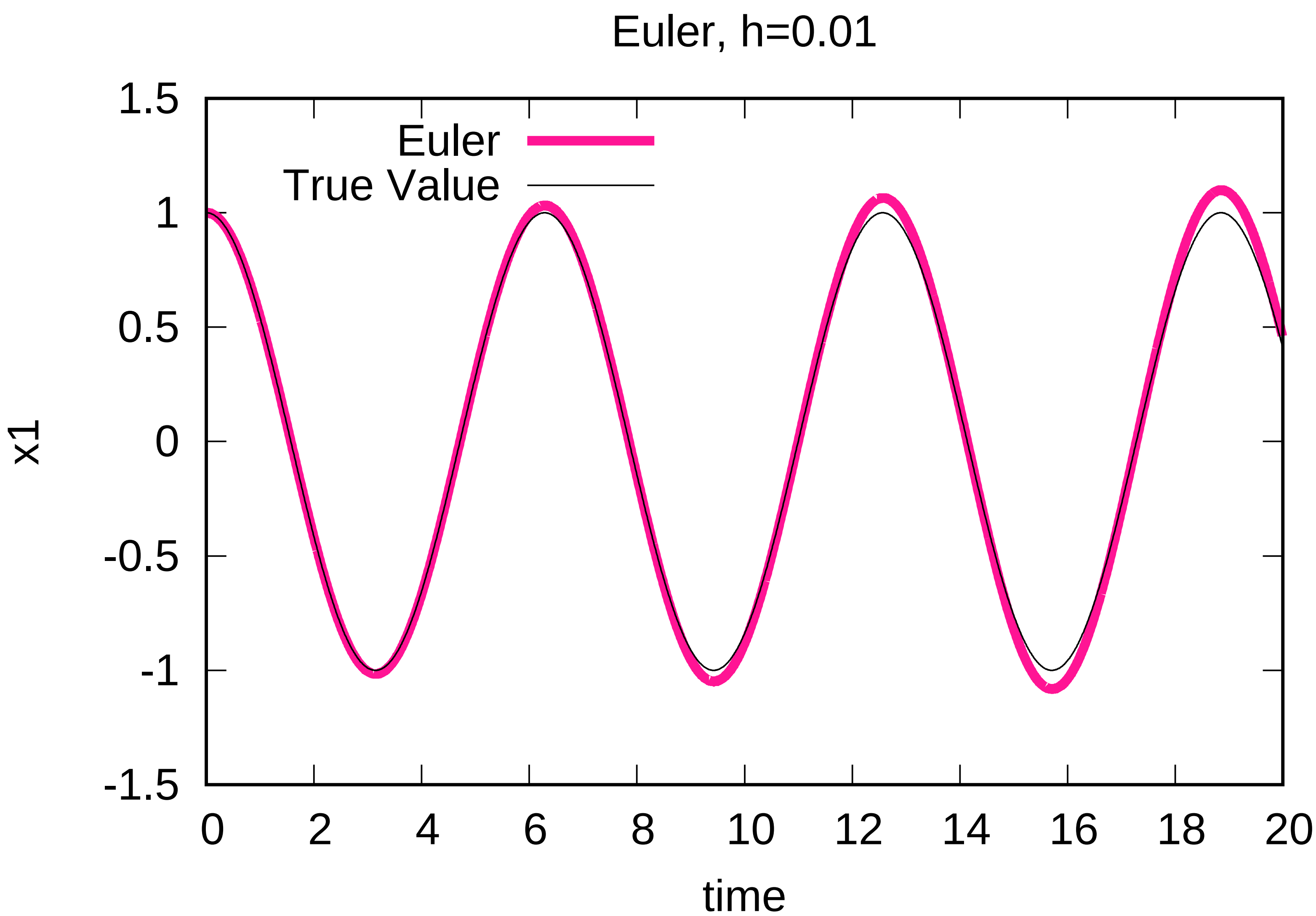
- 前進 Euler 法, 後退 Euler 法, Heun 法による数値解と厳密解を比較すると, 次ページのようになる.



Comparison of Euler, Modifed Euler and Backward Euler
Euler
Heun
Backward Euler
True Value
x1
time
3
2.5
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2.5
0
2
4
6
8
10
12
14
16
18
20

- Heun 法による数値解のグラフは厳密解のグラフと重なっている.
- 前進 Euler 法と後退 Euler 法では, 時間とともに誤差が拡大してゆくが, 誤差が拡大する方向が逆になっている.

- ただし, 前進 Euler 法でも, ステップ幅 `h` を十分小さくすれば, より高い精度の近似解が得られる.
- 前進 Euler 法で `h=0.01` とした場合の数値解を次ページに示す.


Euler, h=0.01
Euler
True Value
x1
time
1.5
1
0.5
0
-0.5
-1
-1.5
0
2
4
6
8
10
12
14
16
18
20