学術フロンティア講義 (S セメスター) 近似を厳密に考える

2. 方程式の解を求めること

齊藤　宣一 (さいとう　のりかず)
数理科学研究科 norikazu@ms.u-tokyo.ac.jp

2019 年 6 月 27 日 (木) 5 限 521 教室

Graduate School of Mathematical Sciences
THE UNIVERSITY OF TOKYO

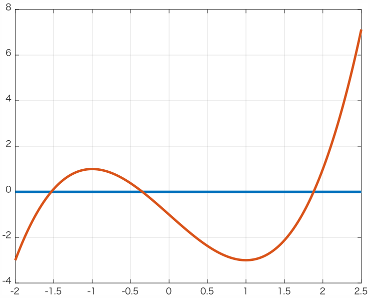
予定

- 6 月 6 日 (齊藤宣一)　1. 積分の値を求めること
- 6 月 13 日 (河澄響矢)
- 6 月 20 日 (河澄響矢)
- 6 月 27 日 (齊藤宣一)　2. 方程式の解を求めること　$\longleftarrow$ 本日
- 7 月 4 日 (齊藤宣一)　3. シミュレーションと多角形の形

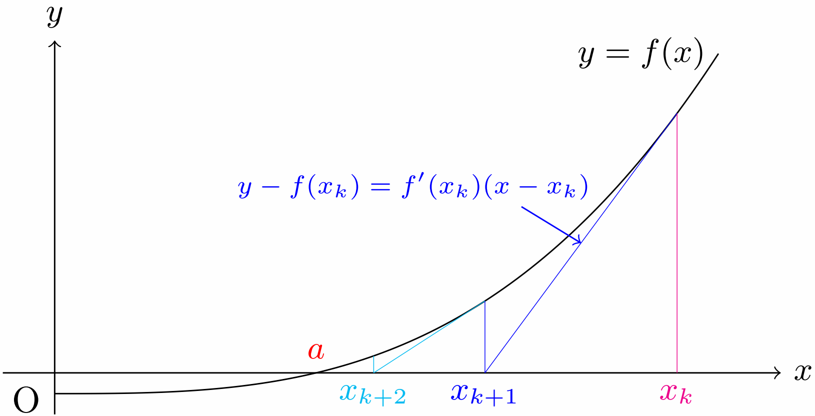
今回のテーマ

関数 $f : \mathbb{R} \to \mathbb{R}$ に対して，方程式 $f(a) = 0$ の解 a を求めたい．

- a を求めるための**解の公式は（一般には）存在しない**．また，解の公式がある場合も，実用的な価値があるかどうかは別問題．
- 一般的な手順としては，反復列 $\{x_0, x_1, x_2, \ldots\}$ で，$x_k \to a$ $(k \to \infty)$ となるものを構成し，十分大きな k に対して，**x_k を a の近似値として採用**する．



代表例：Newton 法



$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

代表的な反復解法

Newton 法

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

$y = f(x)$ の $x = x_k$ における接線 $y - f(x_k) = f'(x_k)(x - a)$ と x 軸との交点を x_{k+1} としている.

簡易 Newton 法

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_0)}$$

この方法では，毎回 $f'(x_k)$ を計算する必要がない.

緩和反復法

$$x_{k+1} = x_k - \lambda_k f(x_k) \qquad (\lambda_k\text{: 定数})$$

Newton 法では $\lambda_k = 1/f'(x_k)$，簡易 Newton 法では $\lambda_k = 1/f'(x_0)$ としている.

収束定理

細かい仮定は省略するが，次が成り立つ．

定理 (局所収束性)

Newton 法と簡易 Newton 法は，求めたい解 a の十分近くに初期値 x_0 をとれば，必ず収束する．

定理 (簡易 Newton 法の収束の速さ)

$g(x) = x - f(x)/f'(x_0)$ とおくと，$\displaystyle \lim_{k\to\infty} \frac{x_{k+1} - a}{x_k - a} = g'(a).$

定理 (Newton 法の収束の速さ)

$g(x) = x - f(x)/f'(x)$ とおくと，$\displaystyle \lim_{k\to\infty} \frac{x_{k+1} - a}{(x_k - a)^2} = \frac{1}{2} g''(a).$

問題 1

どちらの収束が速いだろうか？　理由は?

2 変数の Newton 法

2 元の連立方程式:

$$f(x,y)=0, \quad g(x,y)=0$$

2 変数の Newton 法

$$\begin{pmatrix} x_{k+1} \\ y_{k+1} \end{pmatrix} = \begin{pmatrix} x_k \\ y_k \end{pmatrix} - \begin{pmatrix} f_x(x_k,y_k) & f_y(x_k,y_k) \\ g_x(x_k,y_k) & g_y(x_k,y_k) \end{pmatrix}^{-1} \begin{pmatrix} f(x_k,y_k) \\ g(x_k,y_k) \end{pmatrix} \qquad (\star)$$

実行の際には，逆行列を構成するのではなく，連立一次方程式

$$\begin{pmatrix} f_x(x_k,y_k) & f_y(x_k,y_k) \\ g_x(x_k,y_k) & g_y(x_k,y_k) \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} = \begin{pmatrix} f(x_k,y_k) \\ g(x_k,y_k) \end{pmatrix}$$

を解いてから，

$$\begin{pmatrix} x_{k+1} \\ y_{k+1} \end{pmatrix} = \begin{pmatrix} x_k \\ y_k \end{pmatrix} - \begin{pmatrix} w_1 \\ w_2 \end{pmatrix}$$

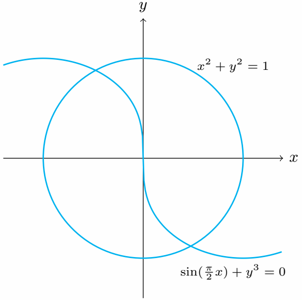
とする．

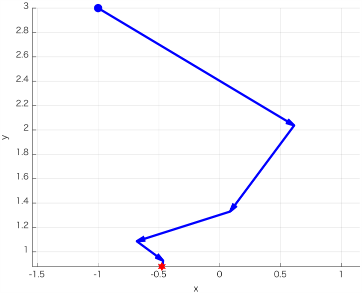
例

連立一次方程式

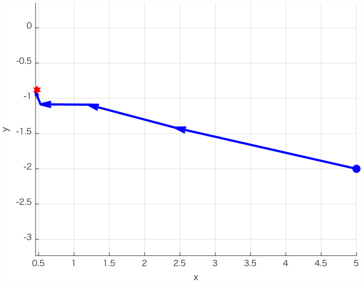
$$f(x,y) = x^2 + y^2 - 1 = 0,$$
$$g(x,y) = \sin\left(\frac{\pi x}{2}\right) + y^3 = 0$$

を解いてみる.





初期値 $(-1, 3)$



初期値 $(5, -2)$**.**

複素 Newton 法

2 元の連立方程式:

$$f(x, y) = 0, \quad g(x, y) = 0$$

2 変数の Newton 法

$$\begin{pmatrix} x_{k+1} \\ y_{k+1} \end{pmatrix} = \begin{pmatrix} x_k \\ y_k \end{pmatrix} - \begin{pmatrix} f_x(x_k, y_k) & f_y(x_k, y_k) \\ g_x(x_k, y_k) & g_y(x_k, y_k) \end{pmatrix}^{-1} \begin{pmatrix} f(x_k, y_k) \\ g(x_k, y_k) \end{pmatrix} \qquad (\star)$$

$z = x + iy$ とおき，連立方程式を次のように書く：

$$u(z) = f(x, y) + ig(x, y) = 0$$

複素 Newton 法　反復列 $z_k = x_k + iy_k$

$$z_{k+1} = z_k - \frac{u(z_k)}{u'(z_k)} \qquad (\star\star)$$

複素微分 (複素関数としての微分)

$$u'(z) = \lim_{h \to 0} \frac{u(z+h) - u(z)}{h} \qquad (z, h \in \mathbb{C})$$

2 変数の Newton 法と複素 Newton 法

定義 (正則関数)

定義域 $D \subset \mathbb{C}$ の全ての点で複素微分可能な複素関数 $u(z)$ を正則関数という.

定理 (Cauchy–Riemann の方程式)

$u(z) = f(x, y) + ig(x, y) : D \to \mathbb{C}$ について次は同値：

(i) $u(z)$ は正則関数

(ii) $f(x, y)$ と $g(x, y)$ が共に (x と y の実数値関数の意味で) 全微分可能であり, Cauchy–Riemann 方程式 $f_x = g_y$, $f_y = -g_x$ を満たす.

定理

$u(z)$ が正則関数ならば, $(\star)$ と $(\star\star)$ は同値である.

注意

f, g が C^1 級であれば, 2 変数のニュートン法 $(\star)$ は適用できる. しかし, Cauchy–Riemann 方程式がないと, 複素ニュートン法 $(\star\star)$ は適用できない.

代数方程式の解法

n 次の代数方程式

$z = x + iy$, $i = \sqrt{-1}$, $c_i \in \mathbb{C}$ $(i = 0, \ldots, n)$ に対して,

$$\begin{aligned} f(z) &= z^n + c_1 z^{n-1} + c_2 z^{n-2} + \cdots + c_{n-1} z + c_n \\ &= (z - a_1)(z - a_2) \cdots (z - a_n) = 0 \end{aligned}$$

の n 個の複素根 $a_1, \ldots, a_n$ を求める.

方法 1. 複素ニュートン法

$$z^{(k+1)} = z^{(k)} - \frac{f(z^{(k)})}{f'(z^{(k)})}$$

の適用が考えられる. ただし, $f'(z)$ は複素変数 z に関する微分を表す. ある近似根 a が求まったら, $f(z)/(z-a)$ に対して, 再び複素ニュートン法を適用すれば良い.

代数方程式の解法

方法 2.

$$f'(a_j) = \prod_{i \neq j} (a_j - a_i)$$

と言う関係に着目しよう．もし，a_j に対する十分良い近似値 z_j が得られているならば，$\prod_{i \neq j} (z_j - z_i)$ は，$f'(a_j)$ の十分良い近似値になっていることが期待できる；

$$a_j \approx z_j \quad \Rightarrow \quad f'(a_j) \approx \prod_{i \neq j} (z_j - z_i)$$

したがって，**すべての根を同時に求める**趣旨で，反復列 $z^{(k)} = (z_1^{(k)}, \dots, z_n^{(k)})$ を，

$$z_j^{(k+1)} = z_j^{(k)} - \frac{f(z_j^{(k)})}{\prod_{i \neq j} (z_j^{(k)} - z_i^{(k)})} \qquad (j = 1, \dots, n) \qquad \text{(DK)}$$

で生成する．これを，**デュラン (Durand)・ケルナー (Kerner) 法**，あるいは，**連立法**と言う．

初期値の選び方

$z^{(0)} = (z_1^{(0)}, \ldots, z_n^{(0)})$ としては，n 個の解 $a_1, \ldots, a_n$ を含むような複素平面上の円盤 $|z-\beta| \leq R$ を構成し，円周上を等分割した点

$$z_j^{(0)} = \beta + R\exp\left[i\left(\frac{2\pi(j-1)}{n} + \gamma\right)\right] \quad (j = 1, \ldots, n)$$

を採用することが多い．γ は適当な実数である．

命題

$f(z) = z^n + c_1 z^{n-1} + c_2 z^{n-2} + \cdots + c_{n-1} z + c_n = 0$ の n 個の根 $a_1, \ldots, a_n$ は，

$$\Gamma = \{z \in \mathbb{C} \mid |z| \leq 1 + \eta\}$$

の中に含まれる．ただし，$\eta = \max_{1 \leq k \leq n} |c_k|$．

注意

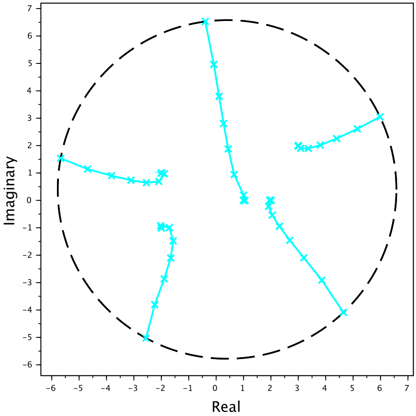
命題の評価はかなり荒いものである．もう少し精密な R と β の導出方法 $\rightarrow$ [齊藤 2017] の §1.6 を見よ

例

方程式

$$f(z) = z^5 - 2(1+i)z^4 - 2(4+i)z^3 + 2(4+5i)z^2 + (31+14i)z - 30 - 20i = 0$$

の根を DK 法で求める．根は $-2 \pm i, 1, 2, 3+2i$ の 5 つである．



基本対称式と根と係数の関係

- n 個の変数 $z_1, \ldots, z_n$
- m 次の基本対称式 $\varphi_m(z_1, \ldots, z_n)$ $(0 \le m \le n)$:

$$\begin{aligned}
\varphi_0(z_1, \ldots, z_n) &= 1; \\
\varphi_1(z_1, \ldots, z_n) &= z_1 + \cdots + z_n; \\
\varphi_m(z_1, \ldots, z_n) &= \sum_{i_1 < i_2 < \cdots < i_m} z_{i_1} z_{i_2} \cdots z_{i_m} \quad (1 < m < n); \\
\varphi_n(z_1, \ldots, z_n) &= z_1 z_2 \cdots z_n.
\end{aligned}$$

$f(z) = 0$ の n 個の根 $\boldsymbol{a} = (a_1, \ldots, a_n)$ は，**根と係数の関係**により，

$$p_i(a_1, \ldots, a_n) \stackrel{\text{def.}}{=} (-1)^i \varphi_i(a_1, \ldots, a_n) - c_i \quad (1 \le i \le n).$$

したがって，$\boldsymbol{a}$ を求めるために，次の連立非線形方程式を考える：

$$\boldsymbol{p}(\boldsymbol{z}) = \begin{pmatrix} p_1(z_1, \ldots, z_n) \\ \vdots \\ p_n(z_1, \ldots, z_n). \end{pmatrix} = \boldsymbol{0}$$

複素連立 Newton 法

複素連立 Newton 法

$$\boldsymbol{z}^{(k+1)} = \boldsymbol{z}^{(k)} - [D\boldsymbol{p}(\boldsymbol{z}^{(k)})]^{-1}\boldsymbol{p}(\boldsymbol{z}^{(k)}) \qquad \left[D\boldsymbol{p}(\boldsymbol{z}) = \left(\frac{\partial p_i}{\partial z_j}(\boldsymbol{z})\right)\right]. \tag{C}$$

命題

複素連立 Newton 法 (C) の各成分は (DK) の形に書ける.

証明. $Q(z) = (z - z_1)\cdots(z - z_n)$ とおくと,

$$\begin{aligned}\frac{\partial Q}{\partial z_j}(z_i) &= \begin{cases} -(z_i - z_1)\cdots(z_i - z_{i-1})(z_i - z_{i+1})\cdots(z_i - z_n) & (i = j) \\ 0 & (i \neq j) \end{cases} \\ &= -Q'(z_i)\delta_{ij}.\end{aligned}$$

ここで，もちろん,

$$Q'(z_i) = (z_i - z_1)\cdots(z_i - z_{i-1})(z_i - z_{i+1})\cdots(z_i - z_n) = \prod_{j\neq i}(z_i - z_j).$$

命題の証明の続き

一方で，$Q(z) = z^n + b_1 z^{n-1} + \cdots + b_{n-1} z + b_n$ と書けば，

$$\frac{\partial Q}{\partial z_j} = \left(\frac{\partial b_1}{\partial z_j}\right) z^{n-1} + \cdots + \left(\frac{\partial b_{n-1}}{\partial z_j}\right) z + \left(\frac{\partial b_n}{\partial z_j}\right).$$

ところで，$b_i = (-1)^i \varphi_i(z_1, \ldots, z_n)$ なので，$p_i(\boldsymbol{z}) = b_i - c_i$，ゆえに，$\frac{\partial b_i}{\partial z_j} = \frac{\partial p_i}{\partial z_j}$.
これらを合わせて，

$$\left(\frac{\partial p_1}{\partial z_j}\right) \frac{z^{n-1}}{-Q'(z_i)} + \cdots + \left(\frac{\partial p_{n-1}}{\partial z_j}\right) \frac{z}{-Q'(z_i)} + \left(\frac{\partial p_n}{\partial z_j}\right) \frac{1}{-Q'(z_i)} = \delta_{ij}.$$

すなわち，$[D\boldsymbol{p}(\boldsymbol{z})]^{-1}$ の第 i 行は，$\left[\frac{z^{n-1}}{-Q'(z_i)}, \ \frac{z}{-Q'(z_i)}, \ \ldots, \ \frac{1}{-Q'(z_i)}\right]$. したがって，$[D\boldsymbol{p}(\boldsymbol{z})]^{-1}\boldsymbol{p}(\boldsymbol{z})$ の第 i 成分は，(便宜上 $b_0 = c_0 = 1$ とおき)

$$\begin{aligned}
&\frac{z^{n-1}}{-Q'(z_i)}(b_1 - c_1) + \cdots + \frac{1}{-Q'(z_i)}(b_n - c_n) \\
&= \frac{z^n}{-Q'(z_i)}(b_0 - c_0) + \frac{z^{n-1}}{-Q'(z_i)}(b_1 - c_1) \cdots + \frac{1}{-Q'(z_i)}(b_n - c_n) \\
&= \frac{-Q(z_i) + f(z_i)}{Q'(z_i)} = \frac{f(z_i)}{Q'(z_i)}. \qquad \square
\end{aligned}$$

Newton 法の由来

- I. Newton (1643–1727)『プリンピキア』(1687) Kepler 方程式

$$x - \alpha \sin x = \beta.$$

- I. Newton (1669) 三次方程式

$$f(x) = x^3 - 2x - 5 = 0 \quad (\text{唯一の実根 } a = 2.09455148\cdots)$$

1. $a = 2 + p$ として方程式へ $\to p^3 + 6p^2 + 10p - 1 = 0$
2. p^2 や p^3 は相対的に微小であると仮定し無視して，$10p - 1 = 0 \to p = 0.1$
3. $p = 0.1 + q$ として方程式へ $\to q^3 + 6.3q^2 + 11.23q + 0.061 = 0$
4. 高次の項を無視して，$11.23q + 0.061 = 0 \to q = -61/11230 = -0.0054\cdots$
5. 結局，近似解として $a = 2 + p + q + r = 2.09455148$ を得る.

この過程は，$x_0 = 2$ として，

$$p = x_1 - x_0 = -f(x_0)/f'(x_0), \qquad q = x_2 - x_1 = -f(x_1)/f'(x_1)$$

などと同値であり，Newton 法そのものである.

Newton 法の由来

- I. Newton (1674 年，手紙) $\sqrt[n]{A}$ ($A > 0$; $n = 2, 3, 4$) を求める方法として，

$$x_{k+1} = [(n-1)x_k + A/x^{n-1}]/n$$

  を提案．これも，$f(x) = x^n - A = 0$ に対する Newton 法そのもの．
- Newton 法は，Newton-Raphson 法とも呼ばれる．J. Raphson (1648–1715) は，

$$x_{k+1} = x_k - f(x_k)/f'(x_k)$$

  の形の反復を系統的に研究した結果をはじめて出版 (1690 年) した人である．(ただし，f や f' という記号は使わず，多項式を具体的に書いていた．) とはいえ，Raphson は，この形の反復法は，Newton からの示唆があったことを述べている．
- Newton 法のアイデアは Banach 空間上の作用素方程式に対しても適用できる．その収束に関しては L. Kantorovich (1912–1986) が基礎的な仕事をして，現在では **Kantorovich の定理**と呼ばれている．この定理は，**非線形偏微分方程式の解の一意存在の証明**で威力を発揮する．すなわち，良い近似法は，良い解析方法となり得るわけである．なお，Kantorovich は数学と経済学の二分野で活躍した人で，1975 年にノーベル経済学賞を受賞している．

復習：Gauss 型積分公式

開区間 (a, b) 上で定義された連続関数 $w(x)$ で,

$$w(x) > 0 \quad (a < x < b), \qquad \int_a^b w(x)\ dx < \infty$$

を満たすものを取り (密度関数)，次の重み付きの定積分の近似解法を考察する.

$$Q_w(f) = \int_a^b f(x)w(x)\ dx$$

定義 (Gauss 型積分公式): $n \geq 1$ とする.

$$Q^{\mathrm{G}}_{w,n}(f) = \sum_{i=0}^{n} W_i f(x_i), \qquad W_i = \int_a^b w(x)L_i(x)\ dx \quad (0 \leq i \leq n).$$

- $a < x_0 < x_1 < \cdots < x_n < b$ は $\phi_{n+1}(x) = 0$ の根
- $\{\phi_k\}_{k\geq 0}$ は，$w(x)$ に対する直交多項式系

復習：性質

定理 I

$q(x)$ を $2n+1$ 次 (以下) の多項式とすると，$Q^{\mathrm{G}}_{w,n}(q) = Q_w(q)$ が成り立つ．
これを，$Q^{\mathrm{G}}_{w,n}$ は $2n+1$ 次精度 の公式であると言う．

- $Q^{\mathrm{G}}_{w,n}$ は，n 次多項式近似に基づいた方法であるにも関わらず，$2n+1$ 次の多項式の積分値を厳密に計算できる．これは，驚くべき性質である．

定理 II

$W_i > 0 \quad (0 \leq i \leq n)$.

定理 III

$\lim_{n\to\infty} Q^{\mathrm{G}}_{w,n}(f) = Q_w(f)$.

疑問と次のテーマ

良い公式だが，$\{x_i\}$ と $\{W_i\}$ をどうやって計算するか？

Gauss 型積分公式の実行

問題設定

区間 $(a, b) = (0, 1)$ 上で重み関数 $w(x) = 1$ を考えて，Gauss 型積分公式の標本点 $\{x_i\}$ と重み $\{W_i\}$ を計算する.

直交多項式

$$
\begin{aligned}
\phi_0(x) &= 1, \\
\phi_1(x) &= 2x - 1, \\
\phi_2(x) &= 6x^2 - 6x + 1, \\
\phi_3(x) &= 20x^3 - 30x^2 + 12x - 1, \ldots
\end{aligned}
$$

$\phi_2(x)$ $(n = 1)$ と $\phi_3(x)$ $(n = 2)$ との場合を目標にしよう.

MATLAB

以下では，具体的な計算プログラムの例を MATLAB で示す.

Durand–Kerner 法

多項式 $\phi_{n+1}(x) = 0$ の根 $x_0, \ldots, x_n$ を求めるために，Durand–Kerner 法を用いる．すなわち，$\boldsymbol{z}^{(k)} = (z_0^{(k)}, \ldots, z_n^{(k)})^{\mathrm{T}}$ を，

$$z_j^{(k+1)} = z_j^{(k)} - \frac{\phi_{n+1}(z_j^{(k)})}{\prod_{j \neq i}(z_i^{(k)} - z_j^{(k)})} \quad (j = 0, \ldots, n)$$

で生成する．このとき，MATLAB では，例えば，

$$\boldsymbol{z} = \begin{pmatrix} z_0 \\ z_1 \\ z_2 \end{pmatrix} \quad \Rightarrow \quad \boldsymbol{z} - \boldsymbol{z}^{\mathrm{T}} = \begin{pmatrix} 0 & z_0 - z_1 & z_0 - z_2 \\ z_1 - z_0 & 0 & z_1 - z_2 \\ z_2 - z_0 & z_2 - z_1 & 0 \end{pmatrix}$$

と計算されることを利用する．すなわち，行列 $\boldsymbol{z}^{(k)} - (\boldsymbol{z}^{(k)})^{\mathrm{T}} + I$ の第 i 行の積がちょうど $\prod_{j \neq i}(z_i^{(k)} - z_j^{(k)})$ となる．MATLAB のコマンドとしては，`prod(A,2)` を利用すれば良い．これは，行列 A の各行の積を成分とする列ベクトルを返すコマンドである．

Durand–Kerner 法のプログラム

Listing 1: dk1.m

```
% Durand-Kerner method
function roots = dk1(func, kmax, tol, xig)
% Newton iteration
x=xig; k=0; dif=tol+1;
len = length(x);
while k<kmax && dif>tol
    f=func(x); df = prod(x-x'+eye(len),2);
    if norm(df,inf)<1.0e-15
        error("df/dx=0");
    end
xnew=x-f./df; dif=norm(f,inf); x=xnew; k=k+1;
end
roots = x;
end
```

重み係数の計算

重み $W_0, \ldots, W_n$ の計算には以下の関係を用いる

$$\begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ x_0 & x_1 & x_2 & \cdots & x_n \\ x_0^2 & x_0^2 & x_0^2 & \cdots & x_n^2 \\ \vdots & & & \ddots & \vdots \\ x_0^n & x_1^n & x_2^n & \cdots & x_n^n \end{pmatrix} \begin{pmatrix} W_0 \\ W_1 \\ W_2 \\ \vdots \\ W_n \end{pmatrix} = \begin{pmatrix} 1 \\ 1/2 \\ 1/3 \\ \vdots \\ 1/(n+1) \end{pmatrix} \qquad (\#)$$

Listing 2: get_weight.m

```
function weights = get_weight(nodes)
len=length(nodes);
weights=fliplr(vander(nodes))'\(1./(1:len)');
end
```

問題 2

$(\#)$ を示せ.

Gauss 型積分公式のプログラム

Listing 3: gauss.m

```
function integ = gauss(func,nodes,weights)
integ = dot(func(nodes), weights);
end
```

Listing 4: ploy2.m

```
% function for Gauss quadrature deg =2
function y = ploy2(x)
    y = (6*x.^2 - 6*x + 1)/6;
end
```

Listing 5: poly3.m

```
% function for Gauss quadrature deg =3
function y = ploy3(x)
    y = (20*x.^3 - 30*x.^2 + 12*x - 1) / 20;
end
```

例題

例題

$g(x) = -x^6 - 4x^5 + 3x^4 + 16x^3 - 11x^2 - 12x + 9$ に対して，

$$Q = \int_0^1 (1-x)\sqrt{g(x)}\, dx = \frac{13}{16}\pi - \frac{23}{15}$$

を計算する．

Listing 6: func45.m

```
%
function y = func45(x)
y = (1-x).*sqrt(-x.^6-4*x.^5+3*x.^4+16*x.^3-11*x.^2-12*x+9);
end
```

例題の計算: $n = 1$, $\phi_2(x)$

```
>> Q5 = 13*pi/16 - 23/15;
>> node2 = dk1(@poly2,100,1.0e-12,rand(2,1))
node2 =
   0.788675134594813
   0.211324865405187
>> weight2 = get_weight(node2)
weight2 =
   0.500000000000000
   0.500000000000000
>> gauss(@func45,node2,weight2)
ans =
   1.028191591450871
>> gauss(@func45,node2,weight2) - Q5
ans =
   0.008980893742497
```

例題の計算: $n = 2$, $\phi_3(x)$

```
>> node3 = dk1(@poly3,100,1.0e-12,rand(3,1))
node3 =
   0.112701665379258
   0.887298334620742
   0.500000000000000
>> weight3 = get_weight(node3)
weight3 =
   0.277777777777777
   0.277777777777777
   0.444444444444445
>> gauss(@func45,node3,weight3)
ans =
   1.019505216742847
>> gauss(@func45,node3,weight3) - Q5
ans =
     2.945190344731952e-04
```

MATLAB について

- この講義では，計算を行う際に，数値計算システム MATLAB

  https://www.u-tokyo.ac.jp/adm/dics/ja/matlabcwl.htm

  を利用した．
- 現在，欧米の数値解析のテキストの計算例は，“ほとんどすべて” MATLAB とを用いて作成されている．プログラム例も MATLAB で書かれている．すなわち，**欧米の大学生は，MATLAB を用いた数値計算の経験をしている**と思って間違いない．

MATLABの利用

東京大学ではMATLABの包括的ライセンスを購入しているので，すべての学生は，教育用計算機システム(ECCS)以外でも，自分のコンピュータでの利用が可能である.

- UTokyo Accountを利用してMathWorksアカウントを取得する必要がある そのためには，“科学の技法・東京大学「初年次ゼミナール理科」副読本(2019年度版)”

  https://fye.c.u-tokyo.ac.jp/students/

  のStep 4.1 MathWorksアカウントの取得とMATLABの利用を参照すると良い.

- ダウンロードの方法も副読本に記されている．オンラインで利用する際には，MathWorksアカウントを取得した後に，

  MATLAB Online https://matlab.mathworks.com/

  にアクセスする.

レポート問題

問題 1 と問題 2 は必須である.

問題 1

p. 6 の定理をもとに，Newton 法と簡易 Newton 法のどちらが収束が速いかを考察せよ.
ヒント：$x_k \to a$ $(k \to \infty)$ と $|x_k - a|$ が十分小さいことを仮定して，p. 6 の定理から何がわかるのかを考える.

問題 2

p. 22 の設定のもとで，$(\#)$ を示せ.

以下の問題 3 は，選択である．意欲のある人は挑戦してください.

問題 3

$\displaystyle\int_0^\pi \sin x \ dx = 2$ を複合 Simpson 則で計算し，収束の速さ (h を変化させたときに誤差がどのように 0 に減衰するか？) を調べよ．p. 27 の例題について同じことを調べよ.